COMPUTER MODELLING OF ADRENAL FUNCTION

A thesis presented for the degree of
Doctor of Philosophy in Electrical Engineering
in the University of Canterbury,
Christchurch, New Zealand.

by

R.B. Jordan B.E. (Hons)

1974

ABSTRACT

The literature pertaining to the modelling of the adrenal-cortical system is reviewed. Two new models describing the behaviour of the adrenal gland under conditions of stimulation from the hormone ACTH are developed. Both are capable of predicting the observed overshoot in cortisol secretion following a step rise in ACTH concentration at the gland. The first relies on depletion of available cholesterol stores in the gland, while the second relies on the proposal of Berger that ACTH stimulates pregnenolone formation and inhibits 17α hydroxyprogesterone formation.

Analysis of measurements of ACTH concentrations in sheep show that real differences between the amount of ACTH entering and leaving the gland do occur. This difference is not due to loss of water or ACTH, nor to transport delays. Therefore, ACTH must be taken up by or otherwise destroyed within adrenal tissue.

Linear models of the inactivation (within the body) of ACTH are studied. It is found that a model with only two compartments explains the differences between bioassay and immunoassay measurements of ACTH concentration. The first compartment contains biologically active ACTH (time constant of breakdown = 12.9 $\pm$ 2.6 min), and the second biologically inactive ACTH which retains immunoactivity (245 $\pm$ 105 min). The model may be used to provide estimates of the concentration of biologically active ACTH from immunoassay data.

A continuous simulation language for the EAI 640 computer is described. This language allows a simulator to study the equations of a small model interactively using a visual display unit and active teletypewriter.

The development of a magnetic tape storage system for the EAI 640 computer is described. The tape system is a backup store for the EAI 260 disc storage system allowing the disc to be kept free for each computer user, and thus allowing larger programs to be executed. The hardware and the software for this system are described.

Between animal variation is not often included in simulation studies. When it is included, it is usually estimated by Monte Carlo techniques, which involve computing large numbers of simulations with the values of the model parameters chosen at random. A method - called the "variant function technique" - is introduced here. The model parameters are replaced by oscillatory functions of time. This permits the statistical behaviour of the model to be estimated from a single extended simulation, for which the computation time needed is strongly dependent upon the number of parameters.

ACKNOWLEDGEMENTS

The continued encouragement and guidance of my joint supervisors Dr R.H.T. Bates, Dr W.S. Metcalf and Dr J.H. Andreae during the course of this project are gratefully acknowledged.

I am equally indebted to Dr E.A. Espiner and Professor D.W. Beaven of the Princess Margaret Hospital Medical Unit and the Christchurch Clinical School, and to members of the Lincoln Group and Princess Margaret Hospital Medical Unit, for their insight and assistance in making this project truly interdisciplinary.

I am indebted to the staff and post graduate students of the Chemistry and Electrical Engineering Departments for assistance in many of the technical aspects of this thesis, and for the many discussions which added stimulation to this research.

To my wife Lynne, who transcribed the original text into a form suitable for typing, and who constantly encouraged me throughout this study, goes my sincere thanks.

During the course of this research I was a grateful recipient of financial assistance (arranged by Dr Espiner and Prof. Beaven) from the North Canterbury Hospital Board.

CONTENTS

PREFACE

The work presented in this thesis was started with the specific aim of using computer modelling techniques to provide a better understanding of the function of the adrenal gland.

As the research proceeded its aims diversified. In chapter 1 the progress of the work and its expansion into varying fields is described.

Chapter 2 is a brief introduction to some relevant aspects of computing, modelling, endocrinology and biochemistry, to acquaint readers from different disciplines with the basic terms and definitions used in these sciences.

A review of the literature relevant to the adrenal cortical system is presented in chapter 3. This covers recent developments in the study of biochemical and physiological aspects of the adrenal system and in the models which have been proposed to represent it.

Chapters 4 to 9 contain the original work presented in this thesis. The first three chapters relate directly to the adrenal system itself, while chapters 7, 8 and 9 describe computational methods and aids which were developed for the purpose of improving modelling techniques.

In chapter 4 two new models of the adrenal cortex are developed and are compared with models which have appeared in the literature.

Experiments made by Espiner, Donald and Hart, in the J.A. Johnstone Memorial Laboratory, Lincoln College, New Zealand, are analysed in chapter 5. In this thesis, it is shown that these experiments provide evidence that the adrenal gland is capable of removing ACTH from the arterial supply, so implying an ACTH receptor in the gland.

Chapter 5 is concerned with the interpretation of radioimmunoassay measurements of ACTH concentration. This is examined further in chapter 6. A model is developed and is tested using results from the aforementioned experiments of Espiner et al. and other published data. A technique for estimating actual concentrations of ACTH from immunoassay results is developed and is applied to two experimental situations. The first is where ACTH is infused into sheep, and the second is where ACTH is secreted by the sheep themselves.

Chapter 7 is concerned with an interactive simulation language called SIMUL8, which was written to enable the models presented in chapters 3, 4 and 6 to be easily analysed on a digital computer.

Chapter 8 describes a magnetic tape data storage system which was built to allow large programs such as SIMUL8 to be used on the EAI 640 computer in the Electrical Engineering Department of the University of Canterbury, Christchurch, New Zealand. The design of both hardware and software are detailed, and an example of the use of the system is presented.

Three methods of incorporating normal biological variations into simulation procedures are examined in chapter 9. One of these methods, which is new, is developed in some detail.

Chapter 10 contains the more important conclusions drawn from the results presented here. Some future areas of research are suggested.

Publications relevant to the research presented herein are:

Espiner E.A., Donald R.A., Hart D.S., Ross Janne, and Jordan R.B., Evidence for adrenocortical uptake of ACTH in vivo. Amer. Jnl Physiol, 226: No. 1, pp. 96 - 104. January 1974.

Jordan R.B. A mass data storage system for a midi computer. Presented at the National Electronic conference, Palmerston North, New Zealand, August 1972.

McKinnon A.E. and Jordan R.B., Stomach emptying modelled on a small computer with oscilloscope display. Presented at the Quantitative Biology Meeting, Nelson, New Zealand, May 1972.

Jordan R.B., Analysis and Interpretation of some Biochemical Measurements using Computer Programmes devised for the purpose. Presented at the New Zealand Endocrinological Society conference, Christchurch, New Zealand, August 1971.

CHAPTER 1

RATIONALE

Prof. D.W. Beaven and Dr E.A. Espiner have been leading a team of endocrinologists (hereinafter referred to as the "Lincoln group") working in New Zealand, partly at the Princess Margaret Hospital in Christchurch, and partly at Lincoln College. They have performed a unique series of experiments on sheep with adrenal transplants. Their work is referenced and reviewed in some detail later in this thesis.

Our initial contacts with the Lincoln group were arranged by Dr W.S. Metcalf, Reader in Chemistry at the University of Canterbury. In collaboration with personnel of the Medical Unit at Princess Margaret Hospital, he has been developing a computer program describing the destination of calcium from dietary and intravenous sources, in the blood, gut and bone of man (cf. Livesey, 1970). Dr Metcalf proposed that a model similar to the calcium program might aid the understanding of, and experimentation on, the complex pituitary-adrenal system. It was with this aim that the research described herein was begun.

During the early part of the research it was found necessary to master the elements of endocrinology, physiology and biochemistry in order to be able to make full use of existing computer modelling techniques. An added bonus was that communication with the endocrinologists was improved.

It became apparent from the study of data collected by the Lincoln group that their experiments were effectively concerned with isolated regions of the pituitary adrenal system. This reflects, not the inadequacy of the experiments, but the complexity of the adrenal system. The available data were insufficient to provide the base for a complete adrenal model. However, much of the data could be used to characterise parts of such a model.

The literature contains a number of models of the adrenal system, all of which are deficient either in their biochemical basis or in their ability to mimic adrenal function. By making use of the most recent experimental results it has been found possible to develop models which have significantly fewer deficiencies than the published models. Two new models of the adrenal cortex have been constructed. They are both different from any that have been published, and are both capable of describing the input-output relationships of the adrenal gland.

The original aim of the research began to be modified after an analysis was made of some experiments designed to measure the amount of the hormone ACTH that was being taken up by the adrenal gland. This analysis casts doubt on the ACTH measuring technique, which appears to give biassed estimates of ACTH concentrations. A model of the processes involved was developed and tested against data generated by the Lincoln group and against other published data. This model, which is one of the main contributions reported in this thesis, allows estimation of true ACTH concentrations in a number of experimental situations.

To undertake computer modelling, one must have access to a computer system which is geared to simulation with regards to both software and hardware. We had ready access to a computer (EAI 590 hybrid computer which incorporated an EAI 640 digital computer and an EAI 580 analogue computer, installed in the Electrical Engineering Department, University of Canterbury), but at the start of our research, little or no simulation of the type of interest to us had been undertaken at Canterbury University. The models investigated during the research were of such variety and form that analogue computation was very time consuming - each model required complete repatching. In order to increase our computing efficiency a digital simulation language was developed in conjunction with Dr A.E. McKinnon, presently at Division of Biological Sciences, National Research Council of Canada, Ottawa. The value of this program is made clear later in this thesis.

To implement such a large program on so small a computer placed serious constraints on the mass storage available. To alleviate this situation a magnetic tape storage system was designed and constructed.

The topics in the penultimate chapter of the thesis (chapter 9), arose from conversations held with a number of endocrinologists and physiologists. Many of these scientists are rather sceptical of modelling for the reason that a model is assigned parameter values which represent those of only one, often hypothetical, member of the species being modelled. Modelling, as currently practised, does not handle normal biological variation. Chapter 9 discusses means of

incorporating normal variations into simulation techniques. One of the methods, developed in conjunction with Dr R.H.T. Bates, is original and has potential advantages, as is shown in chapter 9.

CHAPTER 2

BACKGROUND REVIEW

Science is the observation of facts, and especially of measurable quantities, and the construction of theories that serve to explain and unify such facts and measurements. The theories are tested against whatever facts and measurements can be found or generated by experiments suggested by the theory.

A model is a terse statement of such a theory, and both theory and model are modified or rejected as the discovery of new facts and measurements require. Thus modelling is the process of constructing real or abstract analogues of a system which is to be analysed or described. A model may be little more than a description in words of a system in the simplest case, or at the other extreme may involve large numbers of mathematical equations implemented on a computer. Examples of models are

- the equations of a chemist,
- the molecular models of a biochemist,
- the scale models of a structural engineer,
- the circuits of an electrical engineer, and
- the sketches of an animal physiologist.

Models are sometimes visual descriptions of processes or things which are themselves invisible or partly so. Visual models make the system more easily understood. In this respect models usually contain only the relevant features of the system. Thus, chemical equations, while modelling

the chemical reactions between compounds, ignore the spatial structure of the molecules, which is irrelevant to the stoichiometry.

The more important properties of the models are now listed.

. A model is a terse catalogue of knowledge gained from experimentation, or from analysis of a system.

. A model is usually in a state of change to incorporate new knowledge.

. Models are scaled up or scaled down versions of a real system.

. A model, in practical terms, simplifies reality. System components which are of no interest, need not be incorporated into a model.

. A model transforms system attributes which are difficult to envisage into attributes with which we are familiar, and often into attributes which we have the tools to investigate.

. Modelling should never become an end unto itself. Models are constructed to provoke and to answer questions, and when these questions are answered, the model's function is complete.

In summary of the above assertions, the modelling approach is but one of many tools used in system analysis. It will never replace experimentation, but may certainly aid it.

In the remaining sections of this chapter the forms of models commonly used in studies of physiological systems are described. In particular, computer simulation and its

function in the modelling of biochemical and metabolic processes is discussed.

2.1 Models as Analogues

A model of a system may be represented in numerous analogous ways. Thus the three systems (or models) shown in figure 2.1, are approximate analogues of each other, and, even though they are physically unrelated, they may be thought of as models of each other.

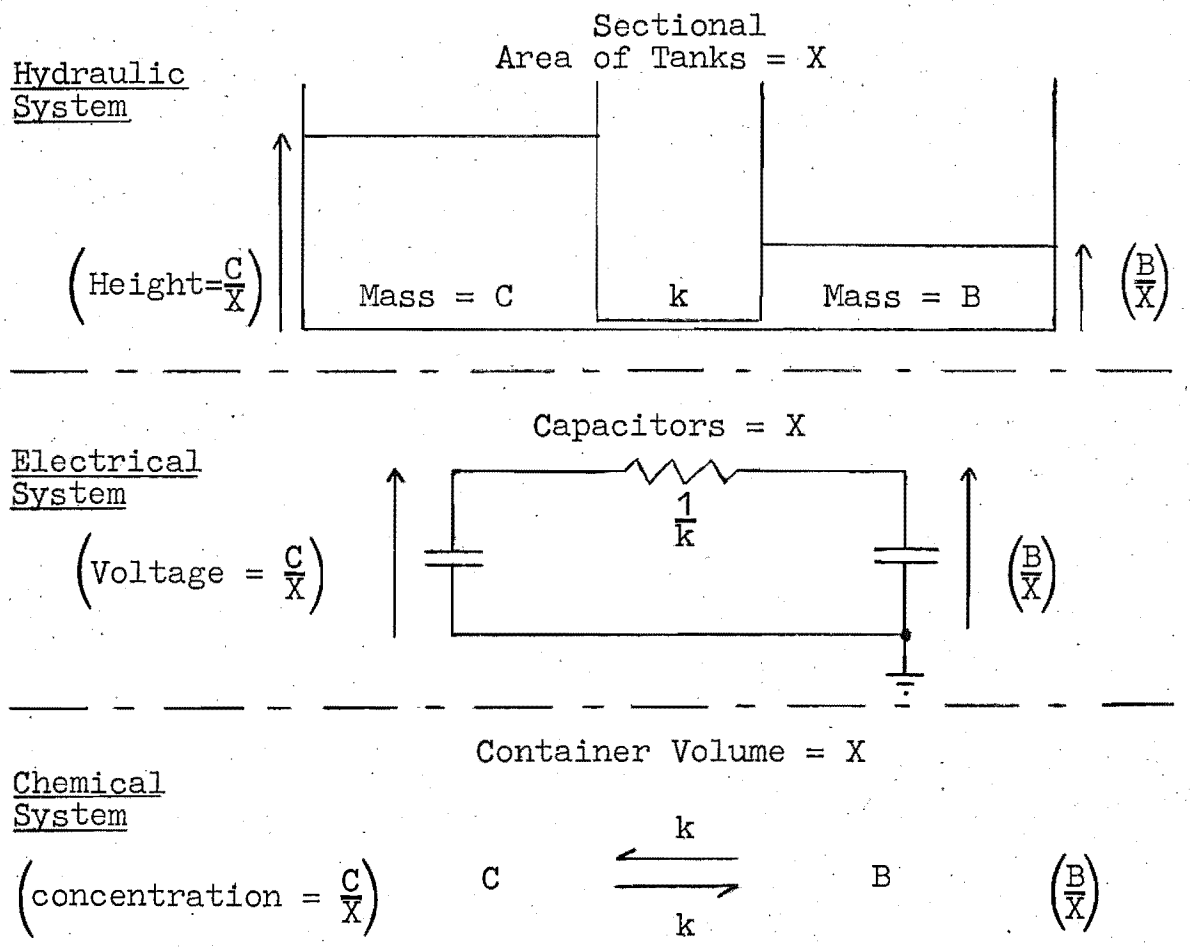


Figure 2.1 Three analogous systems.

That the three systems are in fact analogous is portrayed by writing the differential equations which describe each of them. Thus

$$\frac{dC}{dt} = k\frac{B}{X} - k\frac{C}{X} = -\frac{dB}{dt}\ , \qquad (2.1)$$

describes all three systems in figure 2.1.

Equation (2.1) can be expressed in a form which allows direct implementation on the analogue computer, as in the diagram shown in figure 2.2.

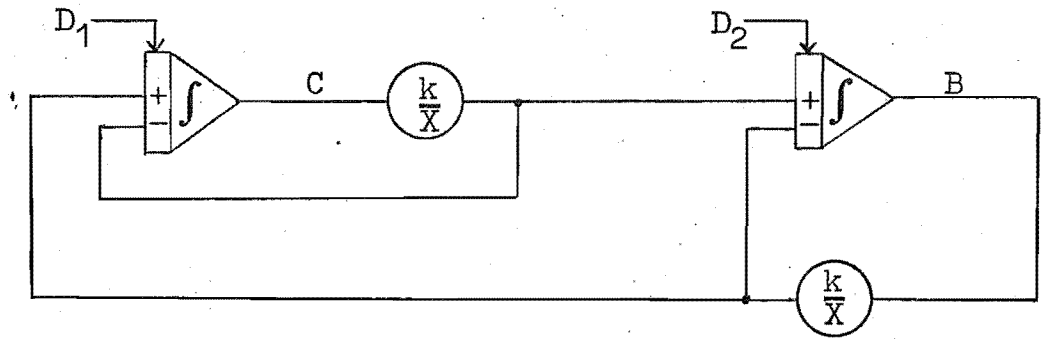


Figure 2.2 Analogue computer representation of equation (2.1).

The equivalence of figure 2.2 and equation (2.1) can be verified by noting that

$$C = \int \frac{k}{X}(B - C)\,dt + D_1\ , \qquad (2.2)$$

and

$$B = \int \frac{k}{X}(C - B)\,dt + D_2\ , \qquad (2.3)$$

where D_1 and D_2 are constants.

It is now seen that figure 2.2 and equation (2.1) are further analogues or models of the system shown in figure 2.1.

All of the models introduced so far may be described using a single diagram with the aid of the compartment notation (cf. Sheppard, 1948). A compartment is a quantity of a substance which has uniform kinetics of transformation and transport (Sheppard, 1948). The three systems shown in figure 2.1 contain two compartments (B and C).

In the conventional compartment notation the compartment size is the driving force of the compartment (e.g. concentration, pressure, voltage), an intensive property of the system. Intensive properties are those which are independent of the amount of the substance or substances within the system (Glasstone and Lewis, 1963, p63). While in general a system is studied by measuring its intensive properties, the mathematics of the system are derived from laws of conservation of "amount" (e.g. mass, energy and charge). The "amount" and the volume in which this amount exists are extensive properties of the system (Glasstone and Lewis, 1963, p63). The ratio of the two is the system driving force.

In this thesis we extend the conventional compartment notation to include both the "amount" and the volume. The conventional compartment size i.e. the driving force, is now expressed as the ratio of the amount to the volume. Thus concentration equals mass over volume, and voltage equals charge over capacitance etc.

The five systems already discussed in this section are summarized, using the new compartment notation, in figure 2.3.

2.2 The Simulation of Systems

The models shown in figure 2.1 allow qualitative analysis of the systems which they represent, but are inconvenient for quantitative analysis in this form. By contrast, the model represented by equation (2.1) is readily analysed

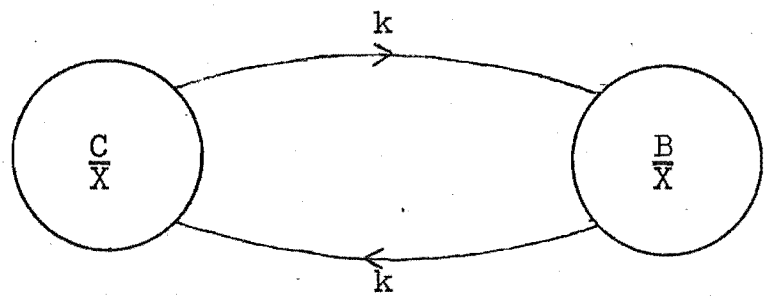


Figure 2.3 The systems shown in figures 2.1 and 2.2, and equation (2.1) represented in the new compartment notation. B and C are "amount" attributes, X the system volume or size and the ratios $\frac{C}{X}$ and $\frac{B}{X}$ the driving forces.

quantitatively, using analytical or numerical methods to yield solutions for each of the model variables.

If the model system under investigation is large, or contains non-linearities, direct algebraic solution of the equations is often tedious or impossible. Such systems are generally solved using analogue computers or alternatively digital computers programmed to behave as analogue computers. The use of computers in this manner is usually referred to as "computer simulation", or as just "simulation". Computer simulation techniques have now progressed to such a degree, that it is often more convenient to solve relatively small systems, which have few or no nonlinearities, using general purpose simulation programs rather than direct algebraic techniques.

On the analogue computer, the equations of the model to be simulated are separated into their component parts (e.g. addition, subtraction, integration etc.), and the components of the analogue computer (e.g. adders, integrators etc.) are

wired together to represent the original equations. This process is illustrated in section 2.1 (cf. equation (2.1) and figure 2.2). Each of the model variables is represented by voltages which change with time in the same manner as do the solutions of the equations. These voltages may be plotted or displayed on an oscilloscope screen automatically to yield visual records of the equation solutions. Analogue computation is potentially very fast because analogue computers operate in parallel, by having a separate analogue component for each operation in the model equations.

Analogue computers have the disadvantage that they are less convenient to set up than digital computers. Before implementation, the equations have to be scaled so that all solutions lie in the range of $\pm$ 10 volts, to prevent "overloading" of the analogue components. When the scaling has been completed the components must be wired together, or "patched". The scaling and patching phases can be time consuming if performed by hand, although programs now exist which can make these processes considerably easier (cf. EAI, 1970).

Because they are generally easier to use and are more readily available than analogue computers, much simulation is now performed on digital computers, often using specially designed simulation languages (e.g. CSMP, Brennan and Silberberg, 1968, DSL/90, Syn and Linebarger, 1966; SAAM, Berman and Weiss).

Simulation languages fall into two broad groups. Firstly there are "discrete" simulation languages which are used to solve difference equations describing the movement

of discrete items (e.g. the components used in an assembly line). Examples of discrete simulation languages are GASP (Pritsker and Kiviat, 1969), GPSS (Herscovitch and Schneider, 1965) and SIMSCRIPT (Kiviat et al., 1968). Discrete simulation languages are not often used in studying continuous systems.

The second group consists of "continuous" simulation languages, such as CSMP (Brennan and Silberberg, 1968) and SIMUL8 (cf. chapter 7). These are used to study models of continuous systems, which are usually based on differential equations. The variables in a continuous system change smoothly from one level to another (e.g. the decay of a radiolabelled chemical in the bloodstream), rather than abruptly as in discrete systems (e.g. the number of bolts in a bin on an assembly line).

The particular technique used to solve the differential equations numerically varies from one language to another, ranging from a simple second order predictor integration such as the Adams-Bashforth method (Bashforth and Adams, 1883) used in SIMUL8 (cf. chapter 7), to high order predictor corrector algorithms such as Milne's method (cf. Noble, 1964), which is used in DSL/90 (Syn and Linebarger, 1966). The selection of the integration algorithm is usually a compromise between speed and accuracy. Lower order algorithms tend to be faster and less accurate than high order algorithms (Gear, 1971).

Two different methods of translating the equations of a continuous system into a simulation language are currently used. The first, which is used in 1130 CSMP (IBM, 1966),

employs a block diagram approach. In 1130 CSMP, the model equations are translated by the user into a block diagram which is similar to that employed for patching the analogue computer. The interconnections between the system components are entered into 1130 CSMP in the form of tables. Block diagram oriented languages are often called analogue computer simulators.

In the second method of presenting the model to the computer, the algebraic and differential equations, coded in FORTRAN-like statements, are entered directly into the computer in the form of a subroutine. DSL/90 (Syn and Lineberger, 1966), 360 CSMP (Brennan and Silberberg, 1968) and SIMUL8 (cf. chapter 7) are all equation oriented languages.

Choice of a language for simulation studies is more often than not dictated by what is available to the simulator. There is no "best" language which suits all people, but rather each simulator has his own preferred language.

Later in this thesis (chapter 7), a simulation language called SIMUL8 is described in more detail. This language has been found convenient to use in investigating metabolic systems such as those described in chapters 3, 4 and 6.

2.3 Modelling Metabolic and Biochemical Systems

Biochemical systems may be investigated by simulating the chemical and transport processes inherent in their operation. A chemical process may be described mathematically

by applying the law of conservation of material or mass (cf. Daniels and Alberty, 1966, p336), and the law of reaction rate (cf. Moore, 1966, p256).

These two laws are illustrated by the following example. Consider the chemical reaction

$$A + B \longrightarrow C \tag{2.4}$$

Conservation of mass requires the total mass of reactants A and B, and product C to be constant. Thus

$$M_A + M_B + M_C = \text{const.} \tag{2.5}$$

Differentiating equation (2.5) with respect to time yields

$$\frac{d}{dt} M_C = - \frac{d}{dt} (M_A + M_B). \tag{2.6}$$

If the volume of the solution is V then

$$[A] = \frac{M_A}{V} ; \quad [B] = \frac{M_B}{V} ; \quad [C] = \frac{M_C}{V} , \tag{2.7}$$

where[A], [B] and[C] are the concentrations of A, B and C respectively. Combining equations (2.6) and (2.7) yields

$$\frac{d}{dt} [C] = - \frac{d}{dt} ([A] + [B]). \tag{2.8}$$

The law of reaction rate relates the rate of product formation to the reactant concentrations. Thus

$$\frac{d}{dt} [C] = k [A][B] = - \frac{d}{dt} ([A] + [B]) , \tag{2.9}$$

where k is the reaction rate constant.

The reaction shown in equation (2.4) is defined as a second order reaction because equation (2.9) involves the product of two concentrations. This should not be confused

with a second order system in control theory which is a system involving a variable and its first and second derivatives.

Often in biochemical systems one of the two reactants is present in abundant quantities which reduces the effective order of the reaction from second order to first order. Thus if

$$\frac{d}{dt}[A] \simeq 0 , \tag{2.10}$$

then

$$\frac{d}{dt}[C] = k'[B] , \tag{2.11}$$

where

$$k' = k[A] . \tag{2.12}$$

All chemical reactions are to some extent reversible (cf. Daniels and Alberty, 1966, p338). That is, the products have a tendency to decompose to form the original reactants. Equation (2.4) would therefore appear as

$$A + B \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C , \tag{2.13}$$

and equation (2.9) is extended to the form

$$\frac{d}{dt}[C] = k_1[A][B] - k_2[C] . \tag{2.14}$$

In many situations the reverse reaction proceeds at such a low rate, that the reaction can be considered irreversible.

Chemical reaction rates are often altered in the presence of catalysts or enzymes, the latter being a catalyst which affects a specific reaction. Catalysts may increase, or may decrease both the forward and reverse rate constants proportionately, and thus they do not affect the equilibrium of the reaction.

The action of a catalyst may be studied using the Michaelis Menten model (cf. Moore, 1966, p313), in which the following reaction sequence occurs

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3} E + P \ , \tag{2.15}$$

where E is an enzyme or catalyst, S a substrate or reactant, P a reaction product and the substance ES is a complex of the enzyme and substrate. The enzyme is not consumed by the reaction and undergoes no chemical change other than the complexing shown in equation (2.15). Thus, the total amount of enzyme present at any stage equals the initial amount of enzyme. In other words

$$[E] + [ES] = [E]_0 \ , \tag{2.16}$$

where $[E]_0$ is the initial enzyme concentration. The mass balance equations for the reactions in equation (2.15) are

$$\frac{d\,[ES]}{dt} = k_1\,[S][E] - (k_2 + k_3)\,[ES] \ , \tag{2.17}$$

$$\frac{d\,[S]}{dt} = k_2\,[ES] - k_1\,[S][E] \ , \tag{2.18}$$

$$\frac{d\,[P]}{dt} = k_3\,[ES] \tag{2.19}$$

After an initial transitory period the concentration of ES reaches a steady value and

$$\frac{d\,[ES]}{dt} = 0 = k_1\,[S][E] - (k_2 + k_3)\,[ES] \,. \qquad (2.20)$$

By eliminating [ES] and [E] from equations (2.16), (2.18), (2.19) and (2.20) the rate of formation of the product P is shown to be

$$\frac{d\,[P]}{dt} = \frac{k_1\,[S][E]_0}{k_2 + k_3 + k_1\,[S]} \,, \qquad (2.21)$$

which has the form

$$\frac{d\,[P]}{dt} = \frac{k_a\,[S]}{1 + k_b\,[S]} = -\frac{d\,[S]}{dt} \qquad (2.22)$$

At low concentrations of S the rate of formation of product is linearly related to the substrate concentration, but when

$$k_b\,[S] \gg 1 \,, \qquad (2.23)$$

then

$$\frac{d\,[P]}{dt} \longrightarrow \frac{k_a}{k_b} = \text{constant}, \qquad (2.24)$$

and the reaction is said to be saturated. Saturating mechanisms such as this are discussed further in chapters 3 and 4.

In certain situations a fifth substance called an inhibitor can interfere with the reaction by forming another

complex with the enzyme, so reducing the amount of enzyme available for product formation. Thus

$$I + E \underset{k_5}{\overset{k_4}{\rightleftharpoons}} EI \ , \tag{2.25}$$

where I is the inhibitor. By using a similar analysis to that above it can be shown that in the presence of an inhibiting agent I, equation (2.22) becomes

$$\frac{d\,[P]}{dt} = \frac{k_a\ [S]}{1 + k_b\,[S] + k_c\,[I]} = -\frac{d\,[S]}{dt} \tag{2.26}$$

It is seen from equation (2.26) that as [I] increases, the rate of product formation decreases, a process called inhibition.

Biochemical transformation often involve consecutive reactions, where a basic molecule has many small chemical changes made to its structure. Such a transformation may be written as

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \xrightarrow{k_3} D \xrightarrow{k_4} E \xrightarrow{k_5} . \tag{2.27}$$

In many situations, some of the rate constants k_1 to k_5 in equation (2.27) are so large that the preceding reactant will be present in negligible quantities, and may be neglected in the mathematical analysis. Thus if k_2 and k_4 are very large when compared to k_1, k_3 and k_5 we may reduce equation (2.27) to

$$A \xrightarrow{k_1} C \xrightarrow{k_3} E \xrightarrow{k_5} \ , \tag{2.28}$$

without affecting the mathematical description of the system.

In a similar vein, if two substances in the sequence are chemically indistinguishable from each other, they may be grouped together for the purposes of analysis with little effect on the observable dynamics of the system. Models involving consecutive reactions, such as those described above, are discussed in section 3.6.

CHAPTER 3

REVIEW OF ADRENAL CORTICAL SYSTEM LITERATURE

The adrenal cortical system consists of the adrenal cortex (which is the outer part of the adrenal gland), the substances it secretes, the bodily processes that these substances control, and a number of other hormones and glands which affect the adrenal secretions. The adrenocortical secretions are essential to life because of their function in controlling systemic concentrations of the minerals sodium and potassium, and the carbohydrate glucose.

Recently, dynamic models of the adrenocortical system have been developed to aid the understanding of, and experimentation on, this complex system.

In this chapter the operation of the adrenocortical system, and the experimental procedures used in its analysis, are summarized. Ten models of the adrenocortical system, which are cited in the literature, are described, and their features compared.

3.1 The Adrenocortical System

Although more than forty different hormones are produced by the adrenal gland, only three (viz. aldosterone, cortisol and corticosterone) are known to be important. Aldosterone controls sodium and potassium concentrations and so is called a mineralocorticoid. Cortisol (more important in humans and sheep), and corticosterone (more important in rats), control glucose concentration and are called

glucocorticoids. All three hormones are formed from cholesterol by the chemical processes of figure 3.1.

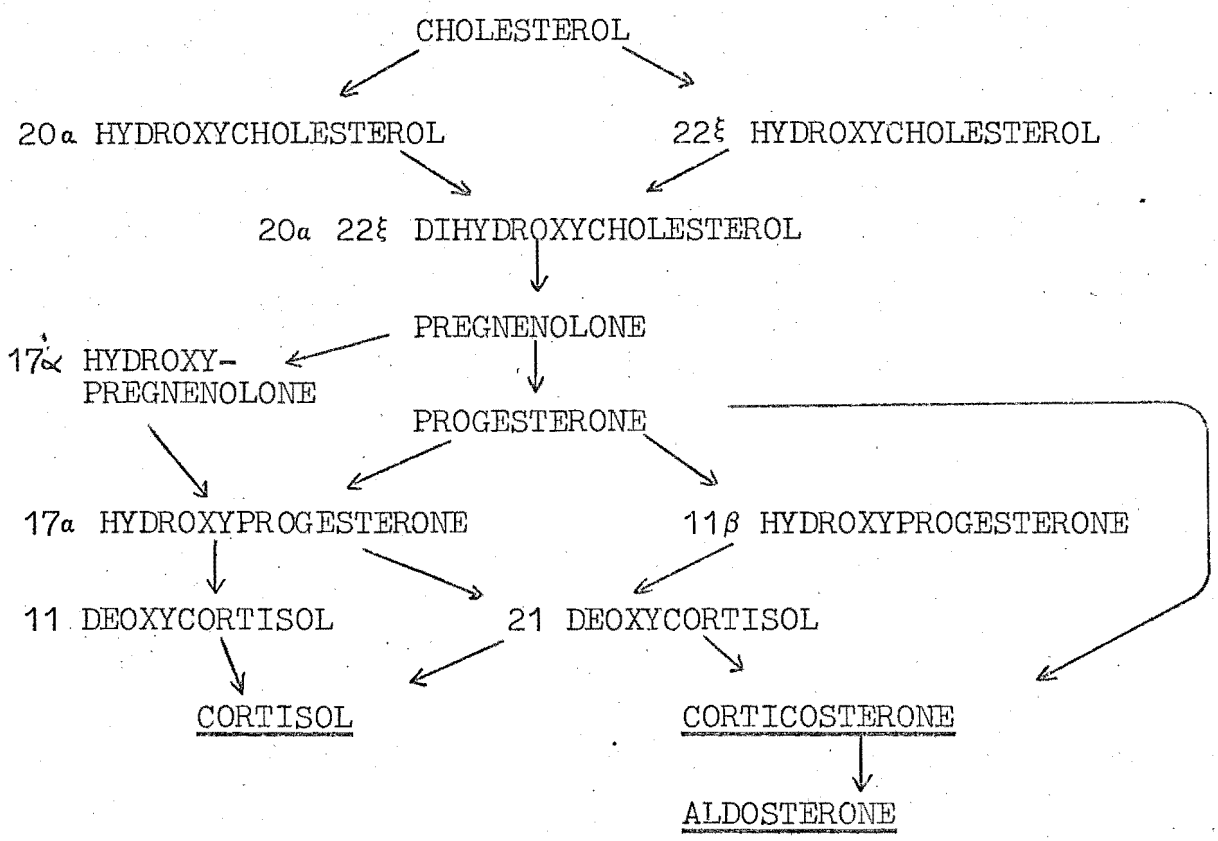


Figure 3.1 The main biosynthetic pathways by which cortisol, corticosterone and aldosterone are formed from cholesterol.

Secretions of the glucocorticoids and mineralocorticoids are controlled by the hormones ACTH (adrenocorticotrophic hormone), and angiotensin II respectively. ACTH is a polypeptide produced in the pituitary. ACTH is in turn controlled by CRH (corticotrophin releasing hormone), a neurohormone produced in the hypothalamus in response to neural stimuli from the midbrain regions. The glucocorticoid control path is shown in figure 3.2. The aldosterone control path, by contrast, does not involve the neural system

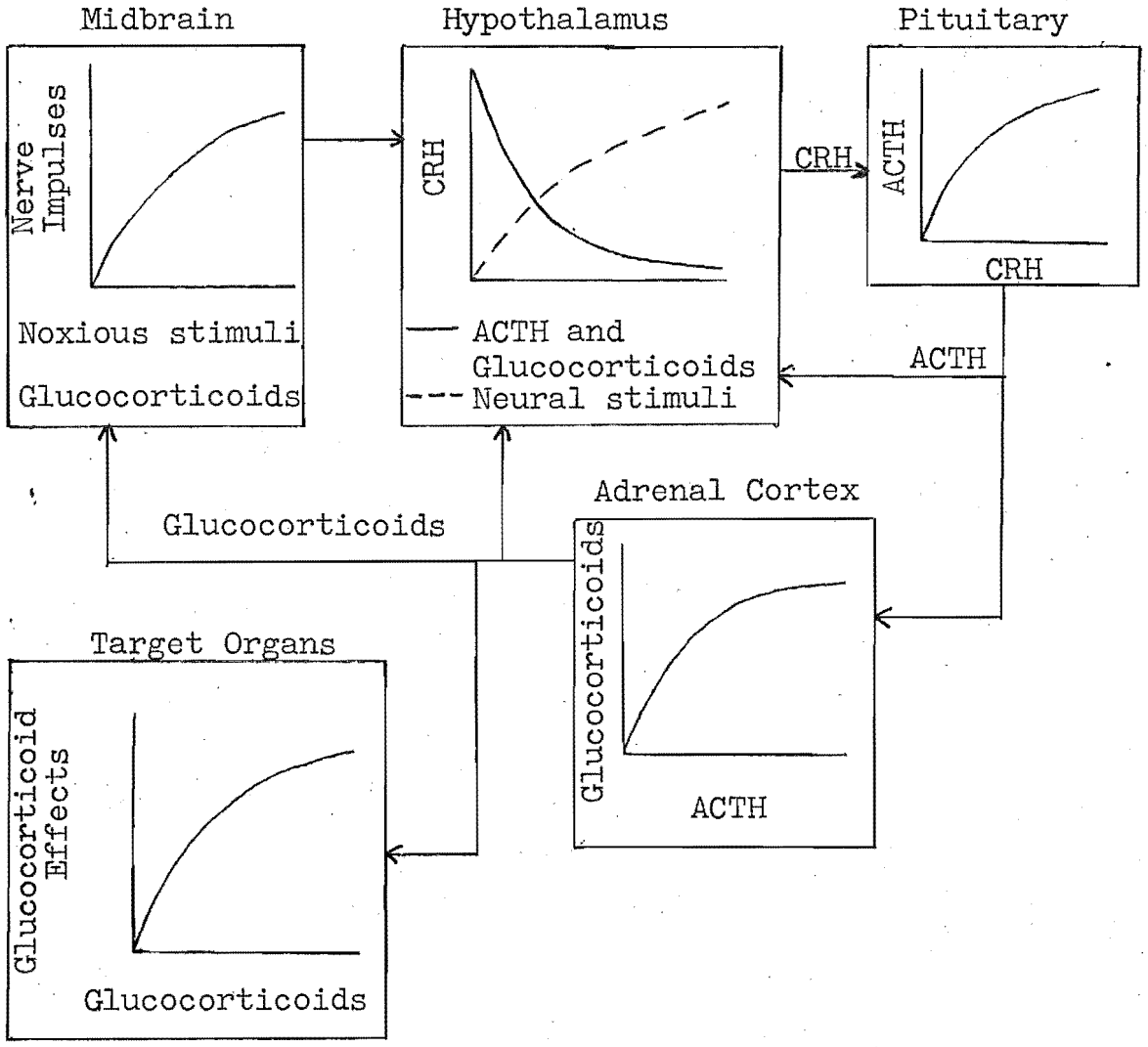


Figure 3.2 Control of cortisol or corticosterone secretion. From Frieden and Lipner (1971).

directly. The path, which is shown in figure 3.3, incorporates parts of the kidney, and enzymes present in the blood. The hormones angiotensin I, angiotensin II, and renin are the intermediates in the aldosterone control loop.

In figures 3.2 and 3.3 the control mechanisms for both the glucocorticoids and the mineralocorticoids are shown as closed loop systems. That is, they contain feedback, which allows the concentrations of the corticoids to be automatically controlled. The evidence for and effect of the closed loop control of the glucocorticoids is discussed in section 3.2.

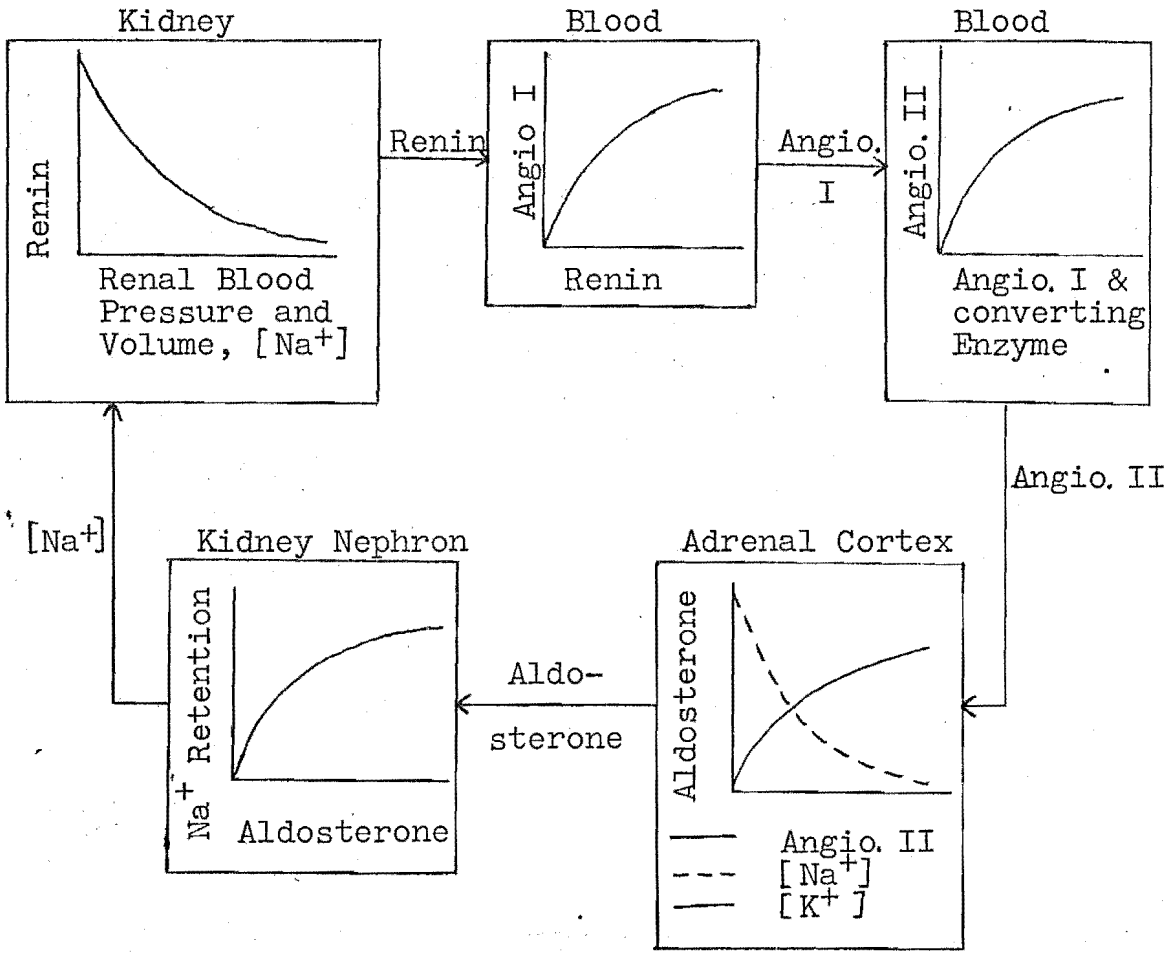


Figure 3.3 Control of aldosterone secretion. From Frieden and Lipner (1971).

The hormones ACTH and angiotensin II do not act independently. For example, angiotensin II has no effect at low concentrations unless ACTH and potassium are present in sufficient amount.

The mechanism by which the hormones ACTH and angiotensin II affect the synthesis of the steroid hormones (i.e. cortisol, corticosterone and aldosterone) is not fully understood. Hall and Young (1968) show that ACTH affects the 20α hydroxylation of cholesterol (cf. figure 3.1). There is also strong evidence to suggest that ACTH does not act directly on this hydroxylation, but controls the formation

of the intermediary cyclic AMP (cyclic 3'5' adenasine monophosphate), within the adrenal cells (Haynes, 1958; Haynes et al., 1959). Figure 3.4 shows the proposed mechanism of ACTH action as summarized by Clegg and Clegg (1969).

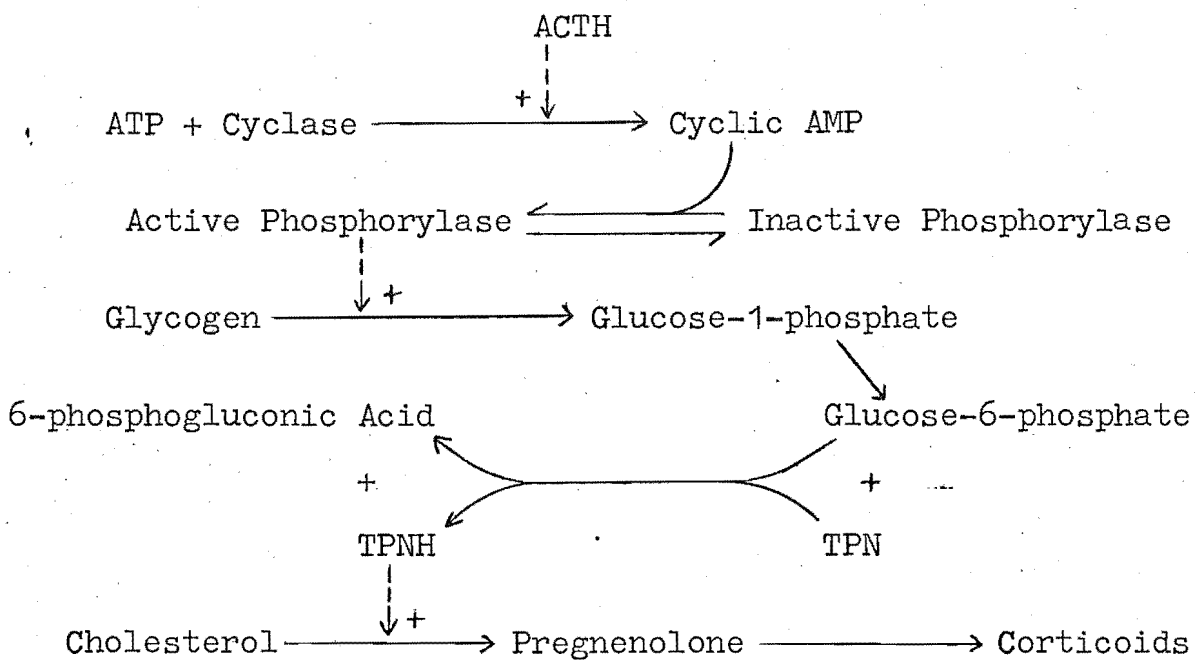


Figure 3.4 Proposed mechanism of ACTH action. From Clegg and Clegg (1969).

3.2 The Glucocorticoid System as a Closed Loop Adaptive Controller

The glucocorticoid system is composed of a number of physically separate parts or subsystems, each with a specific function. Communication between these subsystems is provided by hormones, which act as chemical signals, transported by the bloodstream. The glands of the system have two functions - transduction and amplification. That is they change the signal from one form to another, and at the

same time they increase the signal strength or hormone concentration.

All of the hormonal signals undergo some form of degradation in their movement through the blood vessels. They are diluted and metabolized, and may be bound to proteins and other substances in the blood. All of these degradative processes reduce the amount of the hormone which can act on the "target organ" (i.e. the organ in which a hormone acts, such as the adrenal gland in the case of ACTH).

The glucocorticoid concentrations in the blood are under closed loop control, however, the nature, and the point of action of the feedback, is not clear. That feedback is present is demonstrated by numerous experiments. Espiner et al. (1972 b) inject dexamethasone (a synthetic steroid resembling cortisol in structure and action) into sheep while measuring cortisol secretion rate. Following the injection, cortisol secretion ceases. Conversely, plasma ACTH concentrations are high after adrenals are removed (Hodges and Vernikos, 1960), and where they fail (Bethune et al., 1957). All of these experiments suggest that feedback is present but do not show its form, or where it acts.

Ingle (1959) suggests that some effect of cortisol (e.g. control of glucose synthesis) provides the feedback signal. However, changes in glucose concentration in the blood are slow (Clegg and Clegg, 1969) compared with observed changes in cortisol secretions (Espiner et al., 1972 a),

which seems to exclude glucose concentration as the feedback signal.

Yates and Urquhart (1962) propose from inconclusive evidence that the concentration of unbound cortisol (i.e. cortisol which is not bound to such substances as transcortin or albumin) in the blood provides the feedback.

Frieden and Lipner (1971) report on experiments in which steroid hormones implanted in the midbrain cause adrenal secretion of cortisol to fall. While this suggests that steroids are detected by the neural system, the large doses of steroids used may not be representative of physiological levels of steroids reaching the brain.

There is a strong possibility that there is more than one feedback path. Dallman and Yates (1969) propose that cortisol concentration is detected by two regions of the hypothalamus - the median eminence and septal regions. Also, the presence of a minor control loop is proposed by Kraicer and Conrad (1971). They show that plasma ACTH concentration affects the pituitary secretion of ACTH. The relative importance of these three feedback paths has not been shown, but it is likely that only one or two have predominating effects in the regulation of glucocorticoid synthesis.

Yates and Urquhart put forward a theory for glucocorticoid control which is popularly called the "set point theory". The essence of this theory is that the system is controlled by neural signals which adjust the set point of a simple closed loop controller (cf. figure 3.5). Evidence for such a closed loop system is provided by Yates et al. (1961). They inject separately, and in pairs, the substances

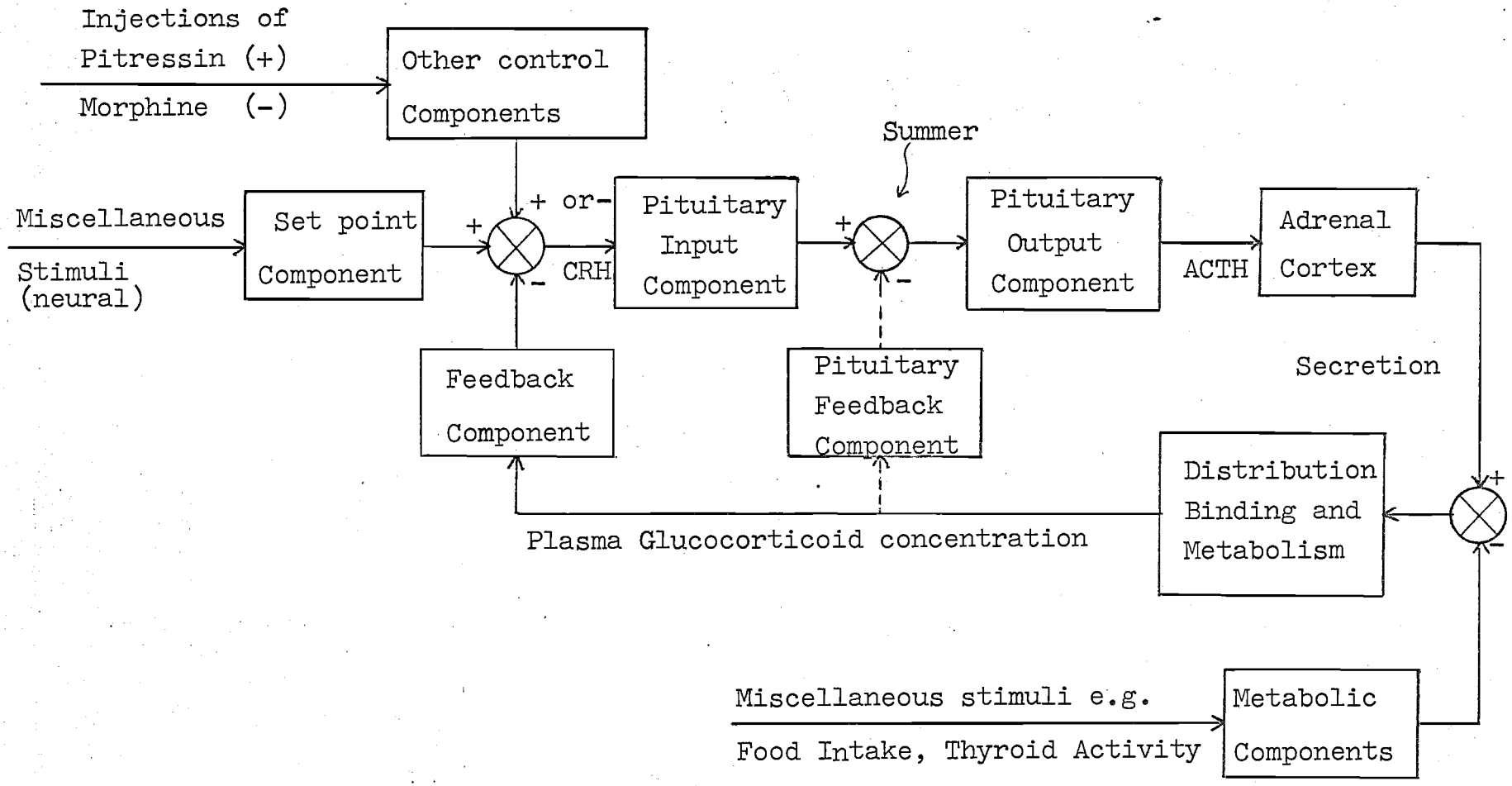


Figure 3.5 The adrenocortical system represented as a closed loop controller. From Yates and Urquhart (1962).

ACTH, corticosterone, and histamine (a noxious stimulant acting in the neural regions which causes the adrenal to increase corticosterone synthesis), into the rat. The amount of each substance injected is adjusted to produce similar increases in the concentration of corticosterone in plasma (cf. figure 3.6). The increase in the concentration of corticosterone, when both ACTH and corticosterone are injected simultaneously, is nearly twice that following

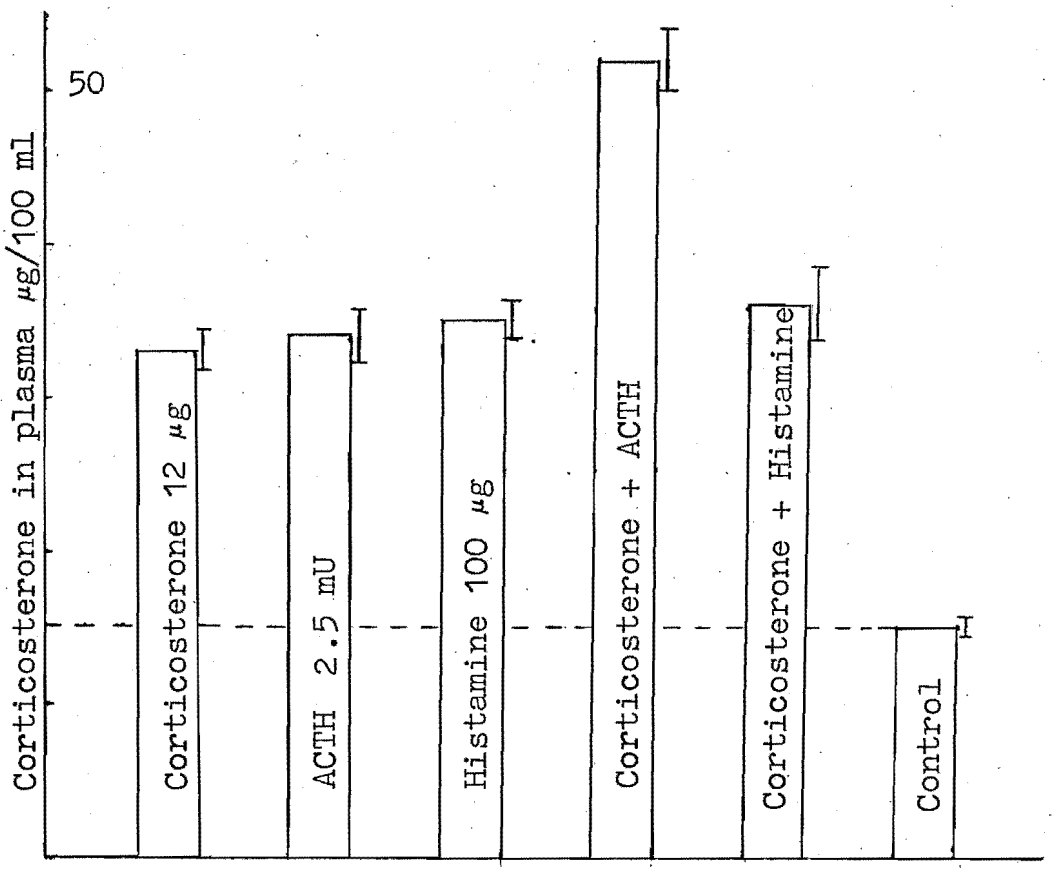


Figure 3.6 Response of rats to various substances injected separately and in pairs. From Yates et al. (1961). ⌶ = ± 1 standard error of the mean.

injection of the two substances separately. In contrast, injection of both histamine and corticosterone does not produce a significant increase in the corticosterone concentration. If the system were an open loop controller,

injection of any two of these substances would produce a concentration increase twice that for one of the substances on its own. With closed loop control, as proposed in figure 3.5, corticosterone and histamine act in opposition at the set point, thus nullifying the effect of the latter substance.

Although the feedback signal in the set point theory is the concentration of glucocorticoids, it is unimportant to the operation of the control loop whether the glucocorticoids, or their effects, provide the feedback; the end result, i.e. control of glucocorticoid effects, will be the same which ever form the feedback takes.

Following continued administration of ACTH, increases in the weight of the adrenal gland are observed. These changes occur over a much longer time scale than the observed changes in glucocorticoid secretion, and therefore do not immediately affect steroidogenesis. Fortier and De Groot (1959) report that in rats, following surgical removal of all but a small fragment of one adrenal, the adrenal fragment doubles in size in about two weeks. The relatively small secretions of the adrenal fragment in this situation are insufficient to maintain normal plasma concentrations of glucocorticoids, causing ACTH to be excreted at increased rates. So that the reduced mass of adrenal tissue can meet the demand placed on it by ACTH, it must increase in size. Such adaptive processes are common in living systems. In this respect the glucocorticoid system may be described as a closed loop adaptive controller.

In this section the glucocorticoid system has been shown as being independent of its environment. In fact this is not so. Many researchers (cf. Yates and Urquhart, 1962) show qualitatively, that the state of the glucocorticoid system is affected by a number of other variables. Bloodflow, the concentrations of steroid cofactors, precursors and enzymes, and the state of closely associated endocrine systems, all affect the operation of the glucocorticoid system and must all be considered when the system is analysed or modelled. For example, insulin, which is secreted by the pancreas, acts in opposition to the glucocorticoids in regulating glucose production. In practice most of these factors are expected never to dominate the control system.

3.3 Adrenal Cortex Experimental Procedures

A number of experimental procedures are used in the analysis of the adrenal cortex, each complementing the others in determining how the gland functions. Three techniques (viz. in vitro studies, perfusion studies and autotransplant studies) are now described.

With in vitro studies the adrenal gland is removed from the animal and maintained in a "live" state by immersion in a solution with properties similar to those of blood. Various hormones and chemicals may then be added and allowed to incubate, or to act on the adrenal cells. Following incubation the chemicals are washed from the tissue and the tissue cells disintegrated by chemical and mechanical means. The component parts of the cell may then be separated by centrifugation, for subsequent chemical analysis. Radio-

chemical labelling of hormones and precursors is commonly used for tracing the biosynthetic paths in the gland in "in vitro" studies. However, care must be taken when interpreting the results from in vitro studies. In extrapolating such results to the in vivo system, the following points must be considered.

. The incubation medium cannot have identical properties to the blood which normally flows through the gland.

. The stimulants added to the tissue medium must be added so that the cells are subjected to similar concentrations to those formed in the in vivo system.

. The blood is continually adding and removing substances, whereas this is not true with in vitro studies, except insofar as diffusion provides transport.

Notwithstanding these points, in vitro experiments have provided much information on the biochemical processes occurring in the glands, much more readily than in vivo experiments provide.

Because of the inaccessibility of the adrenal glands - they are sited on the kidneys - in vivo experiments require surgical techniques to gain access to the glands. The two following techniques are commonly used.

The perfusion technique is used by Urquhart and Keller (1971) for in situ analysis of the adrenal of dogs. Blood drained from the carotid artery of the dog is mixed with perfusion hormones (e.g. ACTH), and pumped into the adrenal artery, which has been disconnected from its normal point of supply. A catheter, inserted in the adrenal vein,

removes sufficient blood, at approximately one minute intervals, for subsequent analysis. The surgical preparation of animals for perfusion studies involves removal of the kidneys and the intestinal blood supply. The pituitary gland is also removed to eliminate the source of endogenous ACTH. Such experiments are costly because of the surgery required, and the fact that the animal must be sacrificed at the end of an experiment. Only input-output relationships between the hormones may be studied, but many of the variables external to the gland, such as blood flow rate, are able to be kept constant. Results obtained from such perfusion studies are discussed in the next section (section 3.4).

The second in vivo technique involves transplanting the adrenal gland from its normal position, to one which is more accessible (cf. Beaven et al., 1964). The right hand adrenal gland is completely removed from the animal, while the left is transplanted to a specially prepared loop of skin in the neck. This loop contains the carotid artery and jugular vein, to which the adrenal gland is joined surgically.

Hormones and other substances may be infused into the gland by pumping them into the carotid artery through a cannula. Blood samples for analysis are siphoned from the jugular vein. The two blood vessels are suitably clamped to ensure that all of the material infused passes through the adrenal, and that little material secreted by the gland can reenter the animals general circulation. This experimental preparation provides good access to the adrenal gland once the initial surgery has been performed. The animals may be reused in experiments over a period of years without

the need for anaesthesia. This technique has most of the advantages of the perfusion technique except for control over blood flow rate.

In conclusion the advantages and disadvantages of the three techniques described are listed in table 3.7.

Table 3.7 Attributes of three experimental techniques for adrenal analysis.

ATTRIBUTE	IN VITRO EXPERIMENTS	PERFUSION STUDIES	AUTOTRANSPLANTED ADRENAL GLAND
Surgery Necessary	Simple	Difficult for each experiment	Difficult prior to first experiment
Animal Sacrificed	Yes	Yes	No
Access to Parts of Cell	Possible	No	No
Access to Parts of Gland	Possible	No	No
Control Over Blood flow	-	Yes	No
Control Over Other Substances in the Blood	Yes	Often Difficult	Often Difficult
Dynamic Studies Possible	Difficult	Yes	Yes
Relationships Found	Intermediate Biochemical Changes and Interrelationships	Input-Output Relationships	Input-Output Relationships

3.4 Results from Dynamic Studies on the Adrenal Cortex

Experiments on the adrenal cortex, in which either the perfusion technique or the autotransplanted gland is used,

allow the dynamic response of the gland to stimulation by ACTH to be measured. In this section a summary of these responses is given.

Following a step increase in the concentration of ACTH, cortisol secretion rises rapidly, reaching a peak in about ten minutes. Subsequently the secretion rate falls slowly reaching a steady level, which is between 60 and 70 per cent of the peak level, in about forty minutes (cf. figure 3.8; Urquhart et al., 1970).

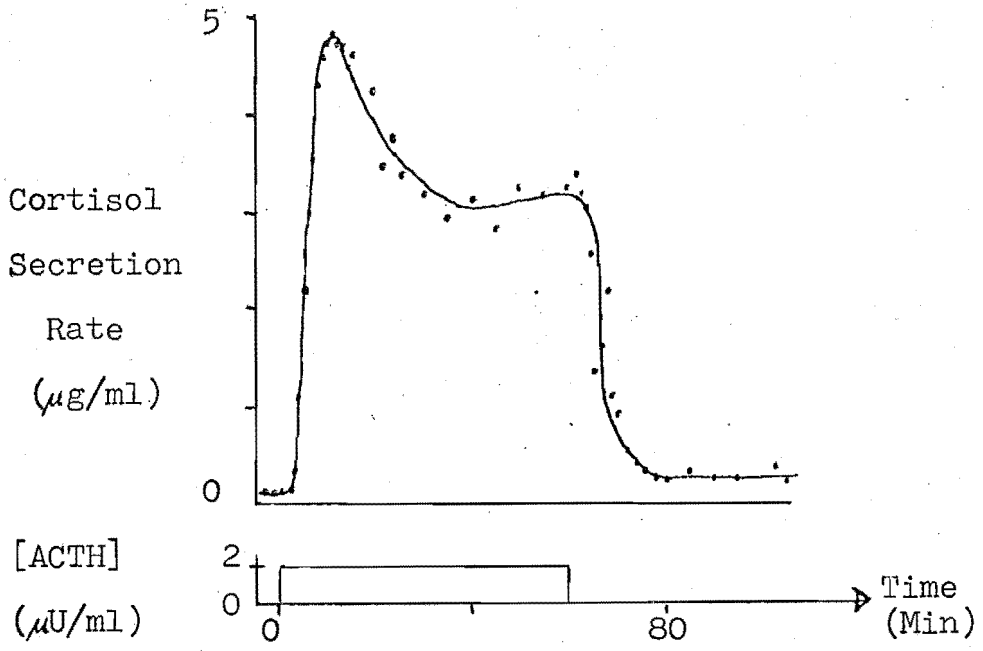


Figure 3.8 Cortisol secretion following a step change in ACTH concentration from 0 to 2 μU/ml. From Urquhart et al. (1970).

Removal of the ACTH causes cortisol secretion to fall monotonically to low levels. The overshoot may be repeated by reapplying the ACTH after a further forty minutes at least. If the ACTH is reapplied after only five minutes, there is little or no overshoot (cf. figure 3.9; Urquhart and Li, 1968).

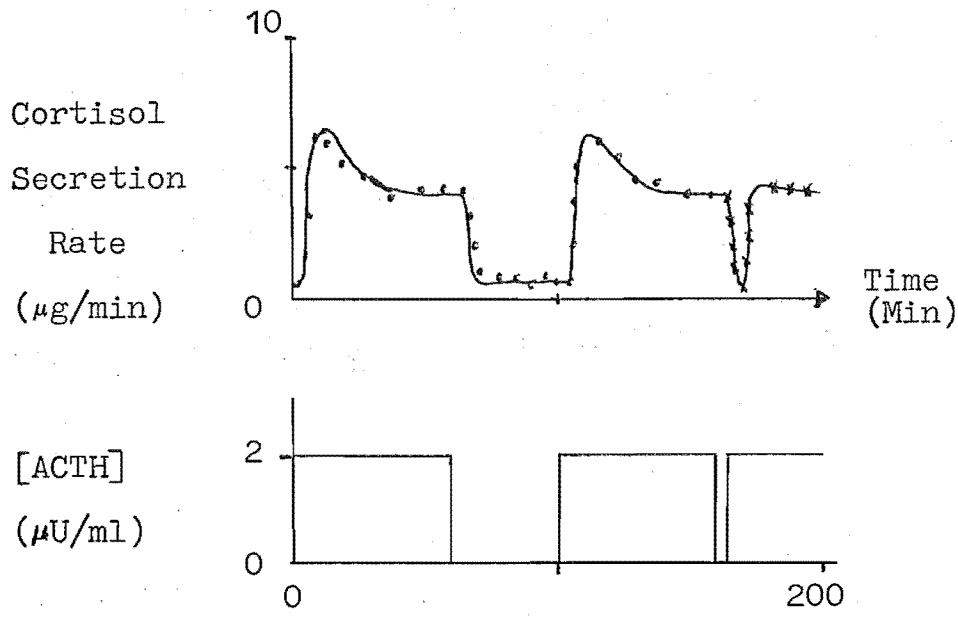


Figure 3.9 Repeatability of overshoot with short and long periods between ACTH infusions. From Urquhart and Li, 1968.

The overshoot fails to appear, also, if very high concentrations of ACTH are infused (cf. figure 3.10). In this case

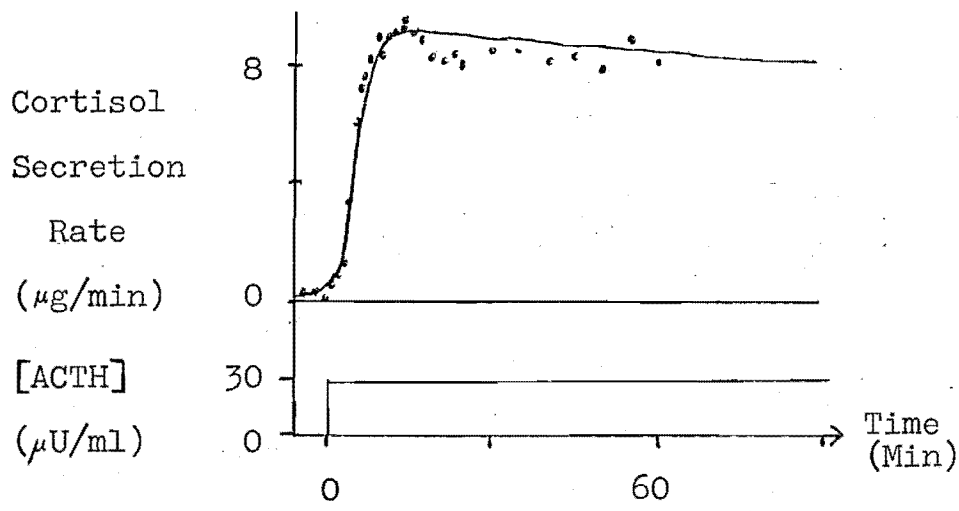


Figure 3.10 Response of adrenal to a high concentration of ACTH. Note that there is little overshoot of cortisol secretion rate. cf. figure 3.8. From Urquhart et al. 1968.

the cortisol secretion reaches a saturation, the form of which is shown in figure 3.11 (Urquhart et al., 1970). Furthermore, when high ACTH concentrations are used, the decline in cortisol secretion when ACTH is removed is further prolonged (cf. figure 3.12; Li and Urquhart, 1969).

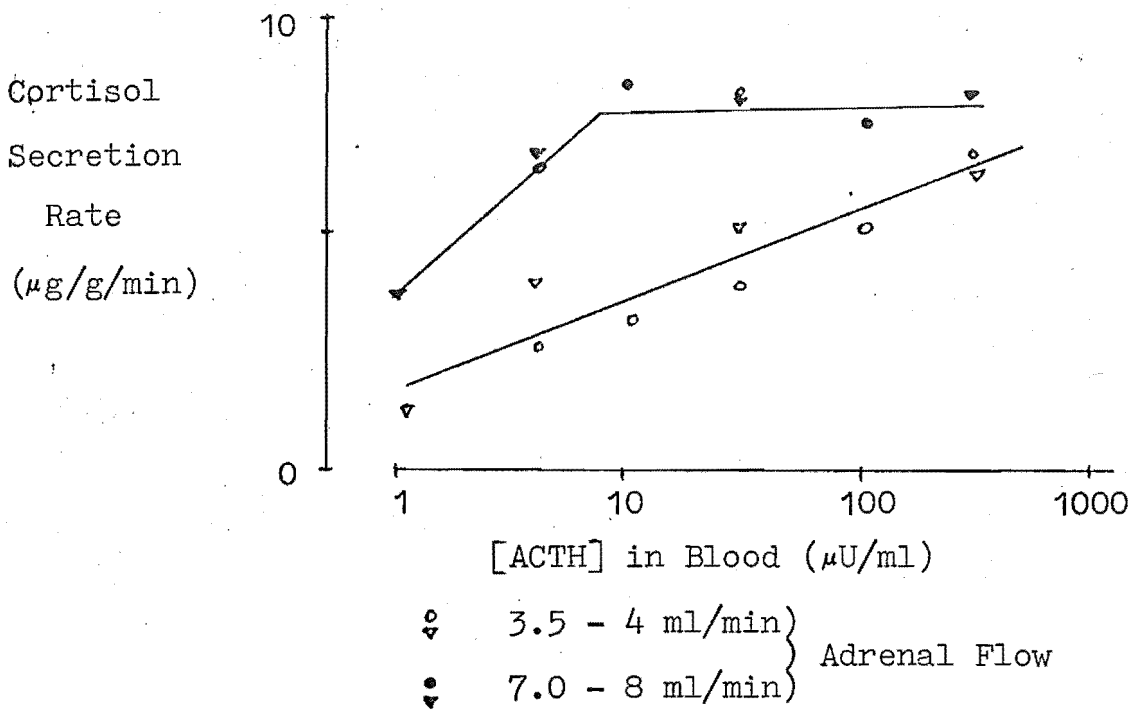


Figure 3.11 Steady state relationships between cortisol secretion rate and ACTH concentration at different blood flow rates. From Urquhart et al., 1970.

The results in figures 3.8 to 3.12, show that the response of the adrenal cortex to ACTH is highly non linear and depends not only on the current concentrations of ACTH in the gland, but also on the amounts of ACTH infused up to forty minutes prior to an observation.

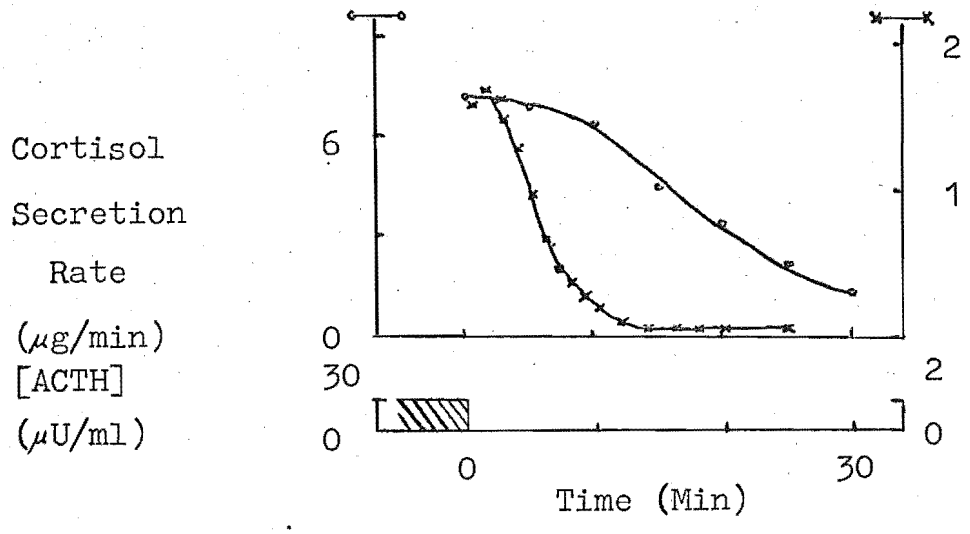


Figure 3.12 Decay of cortisol secretion rate following cessation of ACTH. Two infusions are shown i.e. 30 μU/ml and 2 μU/ml. From Li and Urquhart, 1969.

3.5 Glucocorticoid System Models

The operation of the glucocorticoid system is summarized as follows.

Nervous stimulation of the hypothalamus near the brain produces a secretion (CRH) which excites the pituitary to produce a polypeptide ACTH.

ACTH, as its name adrenocorticotrophic hormone implies, stimulates the adrenal cortex to synthesize a steroid cortisol (or corticosterone) from cholesterol through about a dozen known chemical transformations.

Cortisol is carried by the blood back to the region of the pituitary and hypothalamus where it suppresses the production of ACTH. Both cortisol and ACTH can be estimated chemically, and their interaction is apparent from such measurements.

This is the system which is to be modelled, so that its behaviour as a function of time can be imitated and predicted quantitatively. In the remainder of this chapter, published models of the glucocorticoid system and its component parts are described.

The first dynamic model of the whole glucocorticoid system was published by Urquhart et al. (1959). Their hydraulic analogue of the pituitary-adrenal system shows the main features of this system (cf. figure 3.13). It has more recently been extended and refined to accommodate new data. The operation of the hydraulic model is now described.

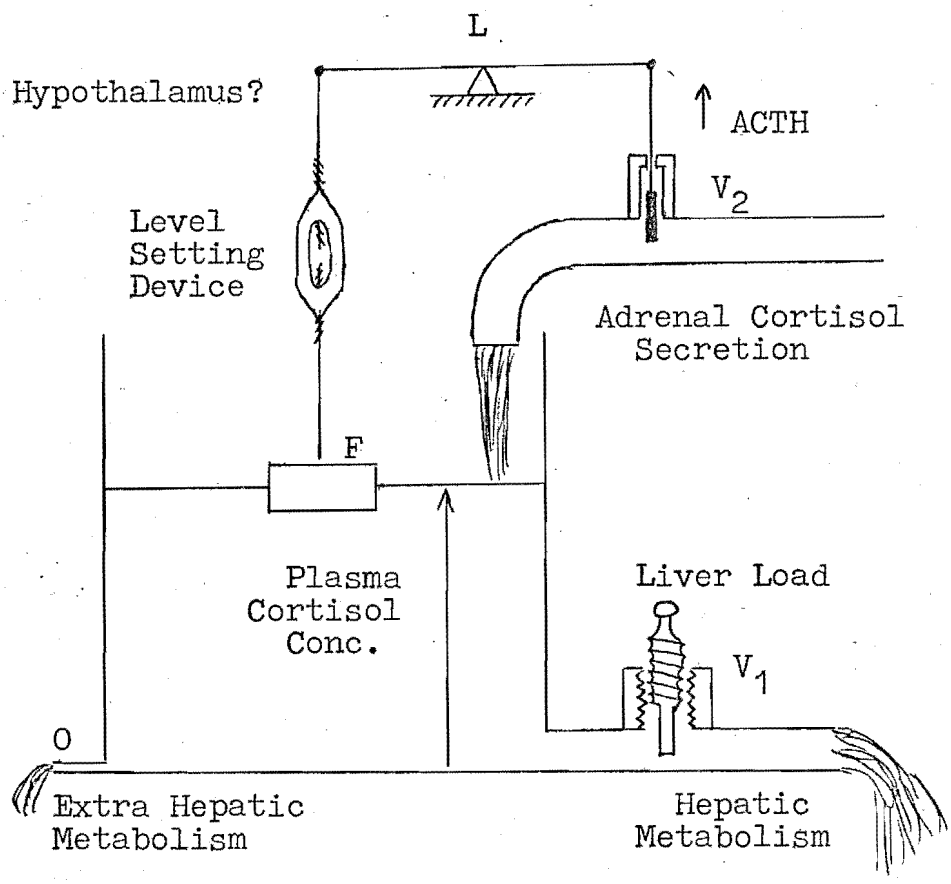


Figure 3.13 Hydraulic analogue of pituitary adrenal system. From Urquhart et al. (1959).

Water in a tank, representing cortisol, is released via two valves. The first is controlled by liver enzymes which degrade cortisol, and the other is fixed, representing other degradative pathways. The flow from both valves is dependent on the height of the water in the tank. Water enters the tank by a third valve, representing the adrenal cortex, which is controlled by a level sensitive device, the hypothalamus and the pituitary, which opens the valve when the water level falls. The action of the model is similar to the common water cistern in which the height of the water is maintained at a steady level.

While the hydraulic model describes most of the important features of the glucocorticoid system, it oversimplifies many of them. Some of these simplifications are listed below.

. Cortisol exists in plasma in a number of forms e.g. free or "native" cortisol, and bound cortisol. It is thought that only the free cortisol affects the feedback site in the hypothalamus (cf. Yates and Urquhart, 1962). In the hydraulic model there is no distinction made between the different forms of cortisol.

. The rate of secretion of cortisol is assumed to be a linear function of ACTH concentration, whereas in reality the relationship between the two hormones is appreciably nonlinear (cf. Li and Urquhart, 1969).

. The control loop is affected by the blood flow rate, the concentration of various substances, and other factors not included in the model.

The model of Dallman and Yates (1969) is similar in function to that of Urquhart et al. (1959), but is described by a block diagram rather than an hydraulic analogue. Dallman and Yates use their model to explain the experimental results described in Section 3.2 (cf. figure 3.6). Two of the variables in the model (shown in figure 3.14), identified by the term "sign restriction" (Dallman and Yates' terminology), are constrained to be non negative.

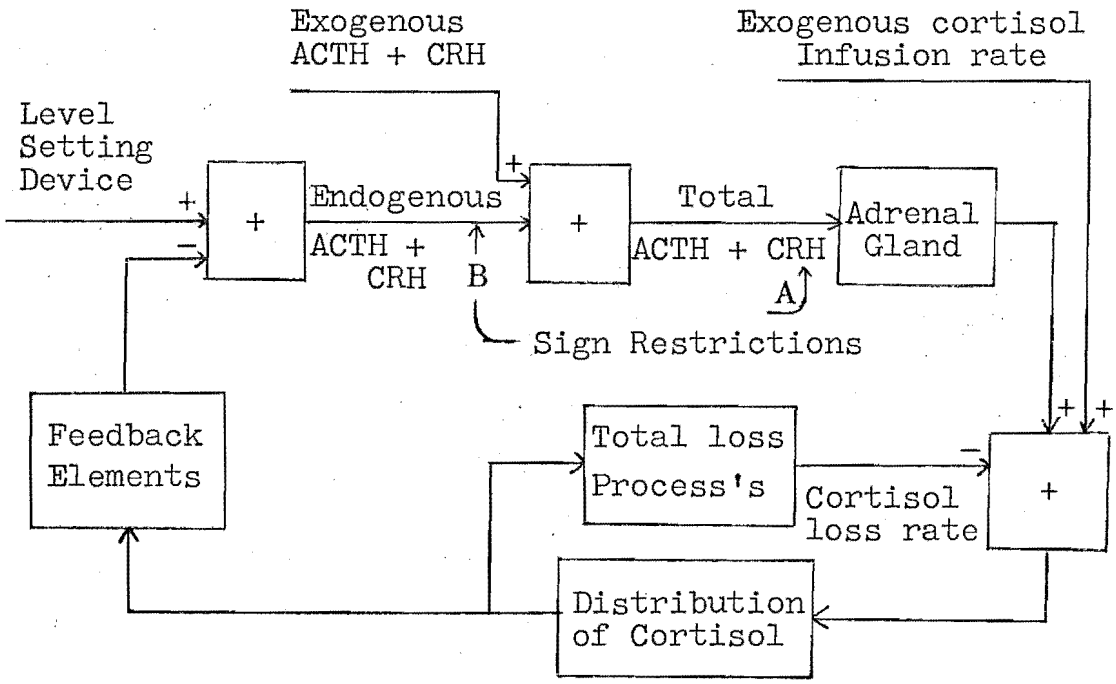


Figure 3.14 Block diagram of pituitary adrenal system. From Dallman and Yates (1969).

These sign restrictions are necessary to prevent the endogenous (sign restriction B) and the endogenous plus exogenous (sign restriction A) concentrations of the hormones CRH or ACTH becoming negative. Analogue computer simulation of the model shows that these sign restrictions must be incorporated to explain the experimental results of Dallman

and Yates (1969; cf. section 3.2). Sign restriction A is unnecessary providing exogenous CRH and ACTH infusions are maintained positive.

The most complex published model of the glucocorticoid system is that due to Yates and Brennan (1967), which is also described by Yates et al. (1968). Their model is reproduced in block diagram form in figure 3.15. Important aspects of the model are listed below.

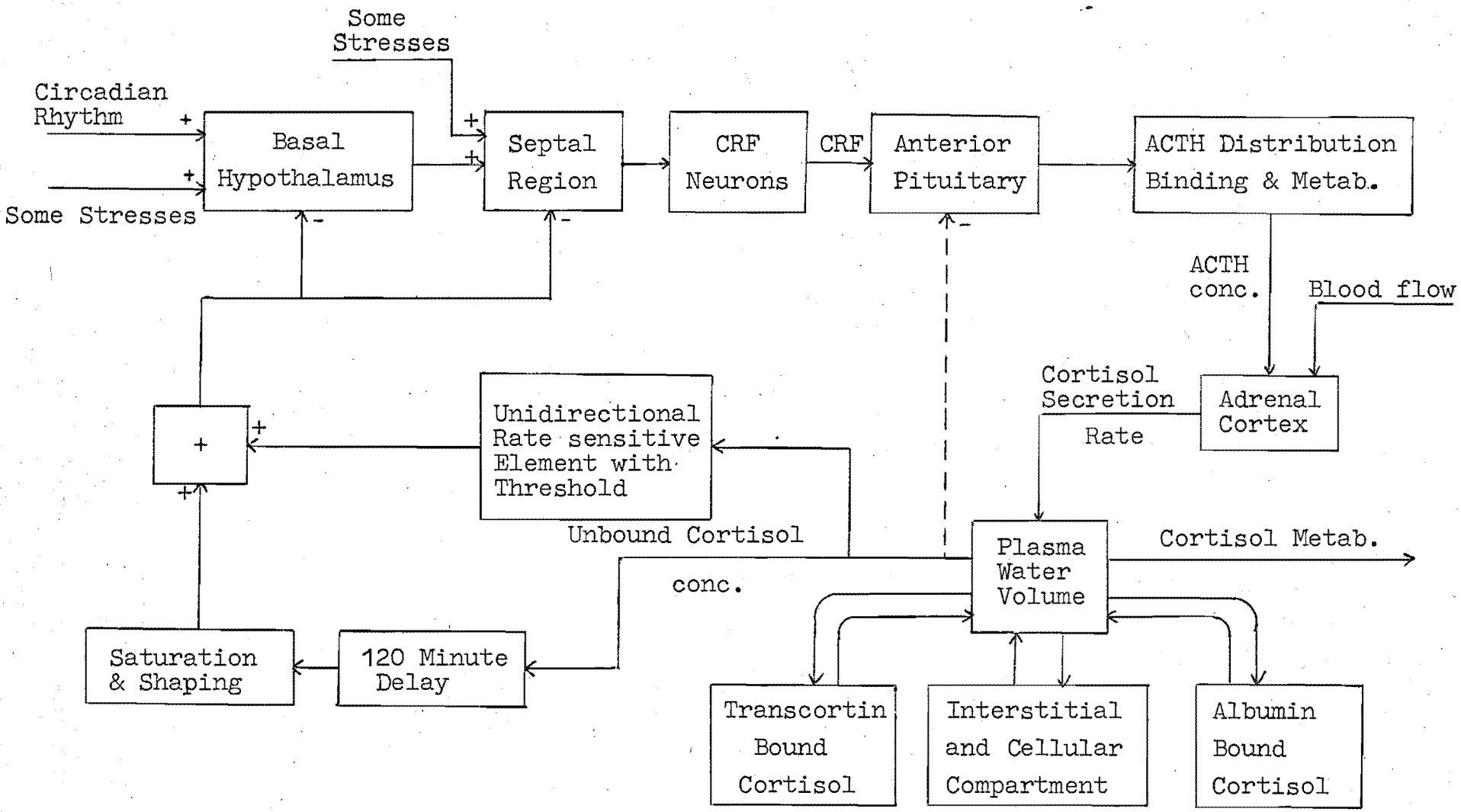
. Three forms of cortisol are described in the model. These are, firstly, free cortisol, which is shown distributed into two compartments - the plasma and the extracellular spaces; secondly, cortisol which is bound to the plasma protein albumin, and finally cortisol bound to a substance called "transcortin".

. The non linearities known to occur in the interaction of ACTH and cortisol, are described by the "Unidirectional rate sensitive" adrenal model which is described in the next section (section 3.7).

. The metabolism of ACTH, its binding to substances in the blood and its distribution in the plasma compartment are accounted for by the model.

. Only the free cortisol affects the feedback path, this feedback acting at two sites in the hypothalamus - the median eminence and septal regions.

With respect to the five points discussed above, the model of Yates and Brennan is significantly more complete than that of Urquhart et al. (1959), which was discussed earlier (p. 38).



3.5 Figure 3.15 Model of the pituitary adrenal system. From Yates and Brennan (1967).

Yates and Brennan (1967) have adjusted the parameters of their model so that the model variables (e.g. the various hormone concentrations) are representative of a small dog. The response of the model, particularly that of the variables cortisol and ACTH, provides a close account of reality. However, some of the model variables are not easily compared with their physical counterparts because the substances cannot be measured. For example, pregnenolone concentration within the cell might be important, but is impossible to measure in the in vivo system.

Many of the glucocorticoid subsystems (e.g. the neural regions), are modelled as functional blocks which provide the right sort of input-output relationships - little attention has been paid to whether the mechanisms of these blocks mimics the real system, simply because so little is known about these regions. Operationally these blocks provide an adequate account of the function of these regions until better information is available.

Empirical mathematical functions are used to describe the input-output relationships of the adrenal gland and hypothalamus-pituitary complex in models developed by Dolkas and Leon (1970), and Stokely and Howard (1972). These models describe the time response of corticosterone and ACTH concentrations in the plasma of rats, following injections of corticosterone. In Stokely and Howards' (1972) model, infusions of CRH (cf. section 3.1) and injections of ACTH may also be programmed. The empirical equations of both Dolkas and Leons' (1970), and Stokely and Howards' (1972) models, are not related to the

biochemical and physiological processes which occur in the glucocorticoid system. In both models, corticosterone and ACTH concentrations are the only model variables with physiological significance.

3.6 Models of the Adrenal Subsystem

The mechanism by which the hormone ACTH stimulates the adrenal cortex to produce cortisol is not fully understood. Numerous theories, derived from in vivo and in vitro studies, have been tested with the aid of dynamic models. In this section the models which have appeared in the literature, and the theories upon which they are based, are described.

Urquhart et al. have published five models of the adrenal system (Urquhart et al., 1968; Urquhart and Li, 1969; Li and Urquhart, 1969; Urquhart et al., 1970). These are now described in chronological order.

The first (Urquhart and Li, 1969) is shown in figure 3.16. It is based on the postulate that ACTH has a catalytic effect

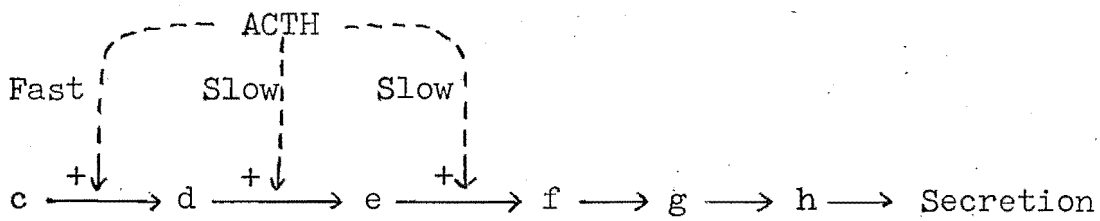


Figure 3.16 The first adrenal model. From Urquhart and Li (1969).

on three biochemical steps in the synthesis of cortisol. The synthetic pathway by which cortisol is formed from cholesterol

(cf. section 3.1), is described by the letters c-h in figure 3.16, each letter representing the quantity of one or more of the biochemical intermediates in the synthesis. When stimulated by small step increases of ACTH concentration, the model shows the biphasic response observed in the real system (cf. section 3.4). However, as the level of ACTH stimulation (i.e. ACTH concentration) of the model increases, the initial rate of cortisol secretion increases much beyond what is found in practice. These high secretion rates could be reduced by introducing a saturation process to one of the later biochemical steps in the model. This is done successfully in the Koritz-Hall model which is described later. As there is little evidence that ACTH acts at three points in the synthetic path, there is no point in continuing the investigation of this model.

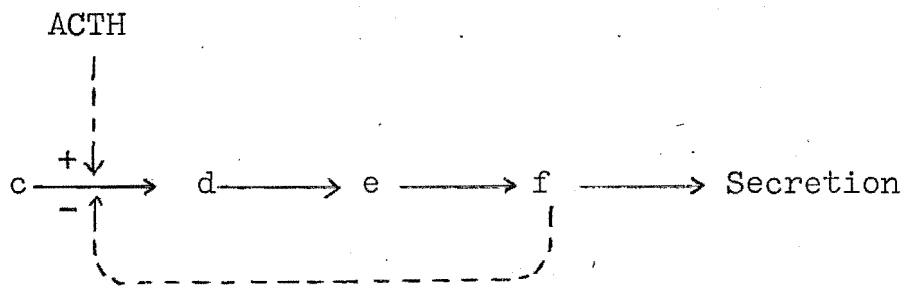


Figure 3.17 The second adrenal model. From Urquhart and Li (1969).

In the second model (Urquhart and Li, 1969), one of the end products (f in figure 3.17) inhibits the reaction (c → d) which is catalysed by ACTH. This feedback mechanism, which is slow to act, reduces the initially high cortisol secretion rate, following a step change in ACTH concentration, thus forming a response similar to that observed in the gland

(cf. section 3.4). Urquhart and Li (1969) reject this model because it fails to reproduce the observed response to sinusoidal changes in ACTH concentration.

The "Unidirectional rate sensitive" model (Urquhart and Li, 1969) which is used in Yates and Brennans' (1967) model of the glucocorticoid system (cf. section 3.6), is the third developed by Urquhart and Li. In this model, two paths in the conversion of cholesterol to cortisol are proposed. One of these (c $\longrightarrow$ c' $\longrightarrow$ d in figure 3.18) provides the initial transient in the response to a step change in ACTH

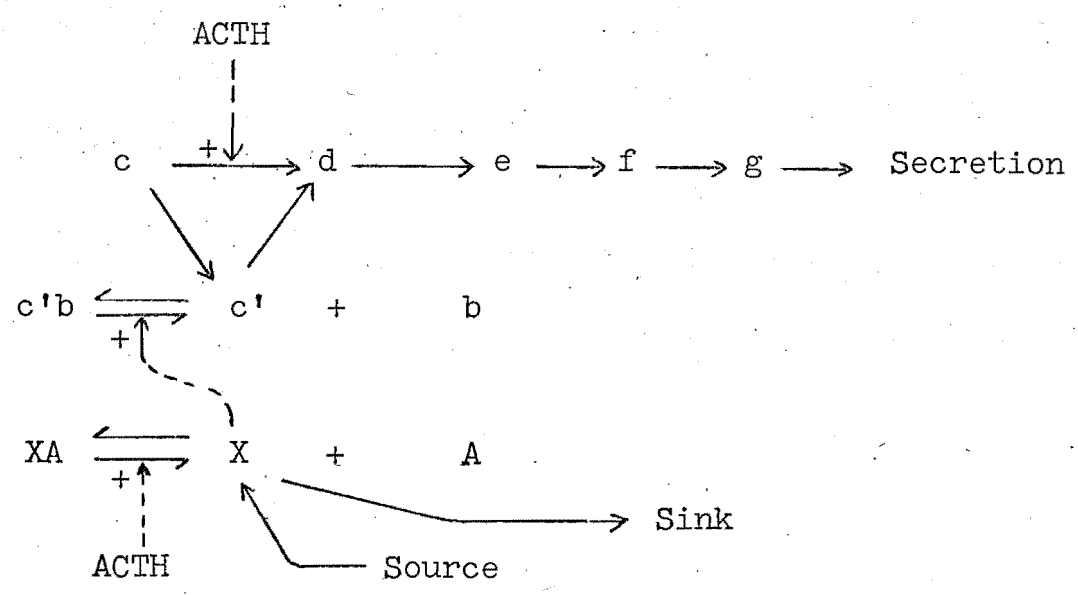


Figure 3.18 The unidirectional rate sensitive model of the adrenal. From Urquhart and Li (1969).

concentration, while the other (c $\longrightarrow$ d), maintains the supply of steroids during the later periods of the response. The transient path relies on the chemical mechanism, shown in figure 3.19 in which an enzyme (I in figure 3.19) affects only the forward rate constant of the chemical reaction $XA \rightleftharpoons X + A$.

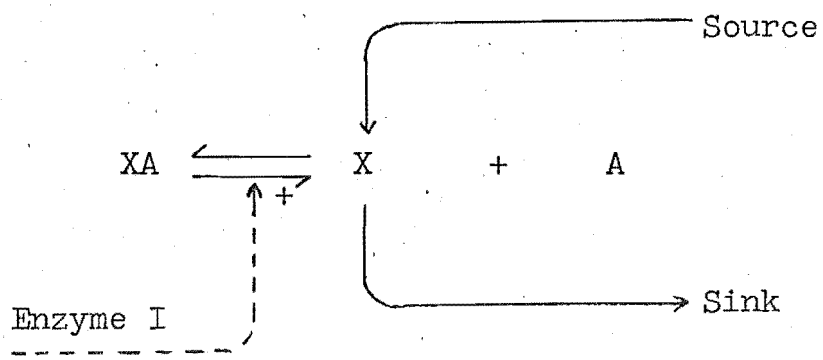


Figure 3.19 Chemical mechanism used in the unidirectional rate sensitive model, which violates the law of conservation of energy.

Catalysts or enzymes, always affect both forward and reverse reaction rate constants equally (cf. Moore, 1966); a catalyst which affects only one reaction path violates the law of conservation of energy, which makes the proposed mechanism questionable. This model, although physically unsound, is very successful in mimicking measured responses of the gland to stimulation by ACTH.

The only model that describes cyclic AMP (cyclic 3'5' adenosine monophosphate) as an intermediate in the action of ACTH, is shown in Li and Urquhart (1969) in sketch form (cf. figure 3.20). Neither the operation of the model nor the equations of this model are described in the text of the paper. In the "second messenger theory" (cf. Gill, 1972; Pearlmutter et al., 1971), cyclic AMP is an intracellular mediator of the action of ACTH and many other pituitary hormones. The fourth model of Urquhart et al. appears to be based on this theory.

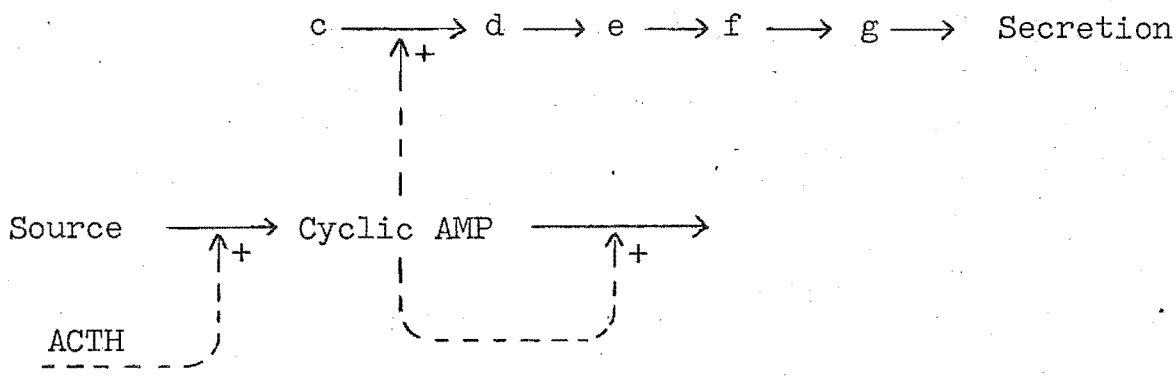


Figure 3.20 The fourth adrenal model. From Li and Urquhart (1969).

The last of the five models developed by Urquhart and Li is the most successful of them all (Urquhart et al., 1968; 1970). The model is developed from a proposal put forward by Koritz and Hall (1964). They report that pregnenolone (an intermediate in the conversion of cholesterol to cortisol) is capable of inhibiting cholesterol hydroxylation (the first chemical change to cholesterol known in the synthesis of cortisol). Pregnenolone, which is formed from cholesterol within the cell mitochondrion (cf. Koritz et al., 1968), must move out of this region before the synthesis of cortisol may proceed (cf. figure 3.2). Koritz and Hall (1964) propose that ACTH might act to increase the permeability of the mitochondrial membrane (which encloses the mitochondrion) to pregnenolone. By this mechanism, in the absence of ACTH, intramitochondrial pregnenolone concentration will increase to a point where it's source is removed by the inhibition of cholesterol hydroxylation. A step rise of ACTH concentration will cause a surge of pregnenolone to move out of the

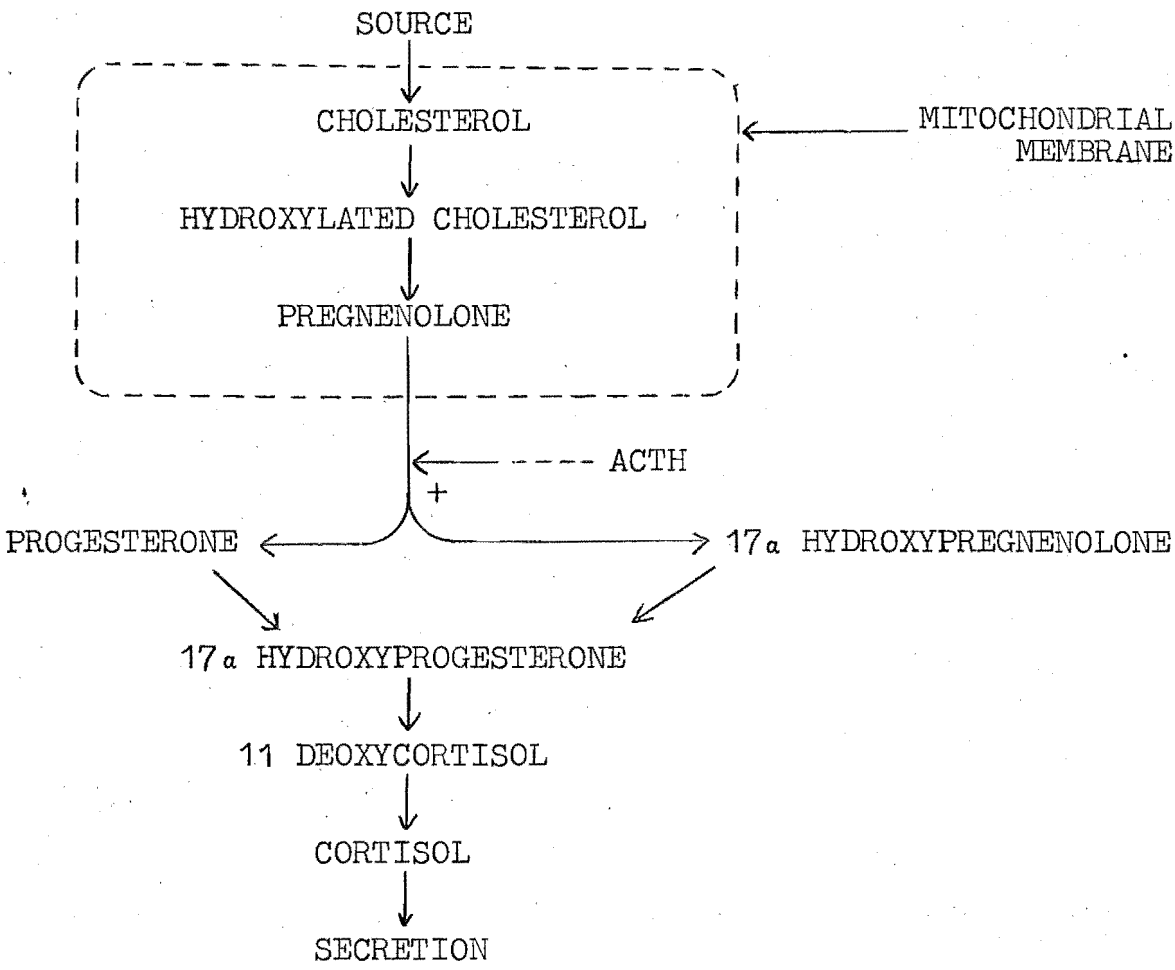


Figure 3.21 Schematic diagram of the Koritz-Hall model of the adrenal. From Urquhart (1970).

mitochondrion, thus causing a rapid initial increase of cortisol production, which gradually diminishes as pregnenolone reserves fall. Loss of pregnenolone from the mitochondrion reduces the inhibition of cholesterol hydroxylation, allowing the synthesis of pregnenolone to proceed. The biochemical reactions describing the process (shown in figure 3.22) are modelled by Urquhart et al. (1968). This model is verified if mitochondrial pregnenolone concentrations fall following stimulation by ACTH. The hypothesis of Koritz and Hall (1964) is based on in vitro experiments. That the in vivo system shows a similar result has not been demonstrated.

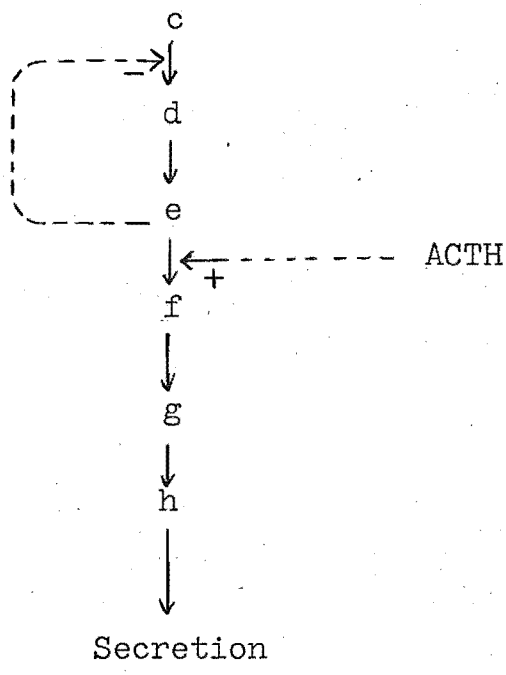


Figure 3.22 Koritz-Hall model of the adrenal. From Urquhart et al. (1968).

The Koritz-Hall model mimics most of the observed gland responses to stimulation by ACTH, and is the model favoured most by Urquhart et al.

Numerous models have been published which show qualitatively the interactions which occur between the parts of the glucocorticoid system. Such models are often used as intermediate steps in the development of dynamic models of the system (cf. Urquhart, 1970a; Dallman and Yates, 1969; Gann et al., 1968; Yates et al., 1968).

Gann et al. (1968) describe a technique for analysing the interrelationships of the hormones and glands which comprise a physiological system. Hormonal signals (cf. section 3.2) are quantized into binary states, allowing

known system interconnections to be described by Boolean equations. Truth tables are formed using experimental data and known interconnections. Further interactions are guessed, and these guesses tested against the truth tables using the rules of Boolean algebra. Gann et al. (1968) claim that this technique provides a useful means of testing the hypotheses made in the modelling process.

In this section numerous models of the function of the adrenal gland have been discussed. The most significant point which can be made about the many models is that they are all widely different in structure. Many theories have been put forward to explain the asymmetric secretion rate of cortisol following step changes of ACTH concentration. All theories are capable of explaining the results as is amply demonstrated by tests made on the models. However, all are light in experimental basis. Nobody, to date has identified the mechanism which causes the asymmetry in cortisol secretion. The models provide the scientist with a means to investigate the overshoot mechanism experimentally. If, for example pregnenolone levels inside the mitochondrion can be shown to rise during ACTH administration, then the Koritz-Hall model is invalidated. In the next chapter further models based on plausible mechanisms of adrenal action, are added to the models which have been described in this section.

CHAPTER 4

SIMPLE ADRENAL MODELS

In the previous chapter (section 3.6) a number of models of the adrenal cortex were described. These models, which are based on theoretical mechanisms thought to occur in the adrenal gland, mimic the experimental results with varying degrees of success. The model of Urquhart et al. (1968), provides the best account of these results, but the Koritz-Hall hypothesis (cf. Koritz and Hall, 1964), upon which it is based, has not been demonstrated by in vivo experiments. Furthermore, recent evidence suggests that ACTH acts prior to pregnenolone in the steroidogenic pathway (Hall and Young, 1968).

In this chapter, two new models of the adrenal system are developed. In the first, the input-output relationships of the gland are analysed, and a mechanism capable of producing such a relationship postulated. The characteristic features of this mechanism are found, and the physiology of the gland investigated to determine where such a mechanism could occur. In the second model, a mechanism which has been proposed to occur in the gland, is tested to see whether it is capable of producing the observed behaviour of the adrenal gland.

The two models developed in this chapter are then compared to the Koritz-Hall model, with regard to their structure, and their ability to mimic the experimental

results. It is shown that there is little to distinguish any of the three models operationally.

4.1 The Depleted Store Model of Steroidogenesis

The characteristics of the adrenals' secretion of cortisol, when the gland is stimulated by step changes of ACTH concentration, are summarized by Yates and Brennan (1967). Briefly these are:

1. a step increase of ACTH concentration in the lower dose range leads to an overshooting response that reaches a peak in about ten minutes. This peak is sixty-five per cent higher than the steady state secretion rate, which is achieved about thirty minutes after the input change.
2. the overshoot is not present if the gland is restimulated within five minutes of the removal of the ACTH.
3. the onset of the secretory response occurs two minutes after a change in ACTH concentration is initiated.
4. at high levels of ACTH stimulation, a point of maximum cortisol secretion is reached.
5. at high levels of ACTH stimulation the decline of cortisol secretion, following removal of the ACTH, is further prolonged.

In view of the characteristics of the adrenal response listed above, the mechanism shown in figure 4.1 is investigated. The mechanism is now described.

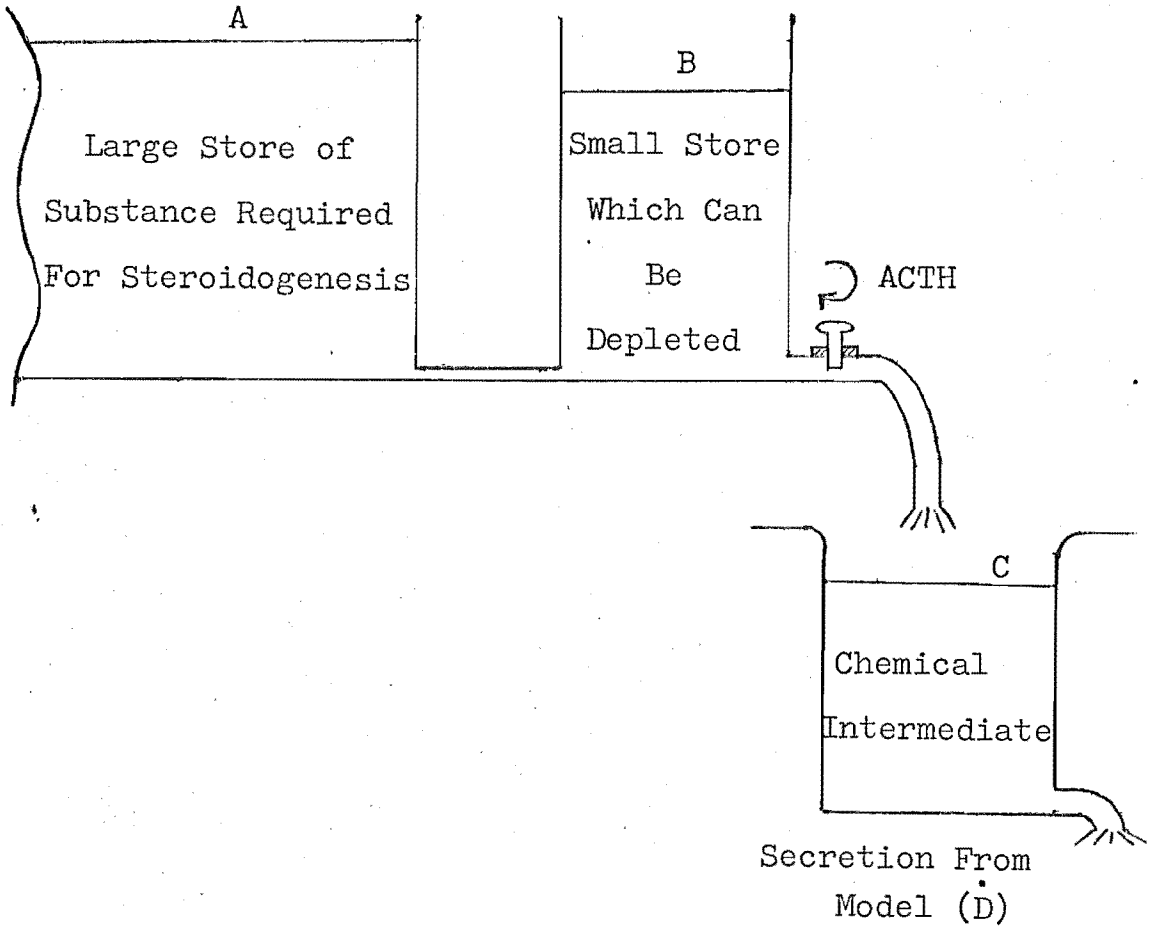


Figure 4.1 Hydraulic analogue of a system which shows a similar response to that of the adrenal gland when stimulated by ACTH.

In the absence of ACTH, the small storage tank B will fill until its height is the same as the large reservoir. When the ACTH operated valve is partially opened, water will flow into tank C at a rate proportional to the height of tank B. From tank C the water will flow away from the system. As the flow from B continues, the height of the water in this tank will drop, which will reduce the outflow rate, thus producing an overshoot in the secretion. If the ACTH operated valve is now closed, secretions from C will rapidly cease, allowing tank B to slowly refill. Only if

the tank is allowed to refill completely will the overshoot be repeated by the reapplication of ACTH.

At high levels of ACTH a different mechanism comes into effect. The height of the water in tank C will rise to the point where the diameter of the tank increases. At this point it will require a large amount of water to appreciably increase the height of water in the tank. Consequently the flow out of tank C will saturate. Furthermore, when ACTH is removed the extra water will take longer to flow out, thus delaying the decline of secretion from the system.

The equations of the model just described are

$$\dot{A} = 0 , \tag{4.1}$$

$$\dot{B} = k_1 A - k_2 B - k_3 B\,[ACTH] , \tag{4.2}$$

$$\dot{C} = k_3 B\,[ACTH] - k_4 C/(1 + k_5 C) , \tag{4.3}$$

$$\dot{D} = k_4 C/(1 + k_5 C) . \tag{4.4}$$

The solutions of equations (4.1) to (4.4), under stimulation by varying concentrations, and repeated infusions of ACTH are shown in figures 4.2 and 4.3. The parameter values used in the simulation, which is performed using SIMUL8 (cf. chapter 7), are

$$A = 200 , \quad k_1 = 0.05, \quad k_2 = 0.05,$$
$$k_3 = 0.025, \quad k_4 = 1.32, \quad k_5 = 0.154 . \tag{4.5}$$

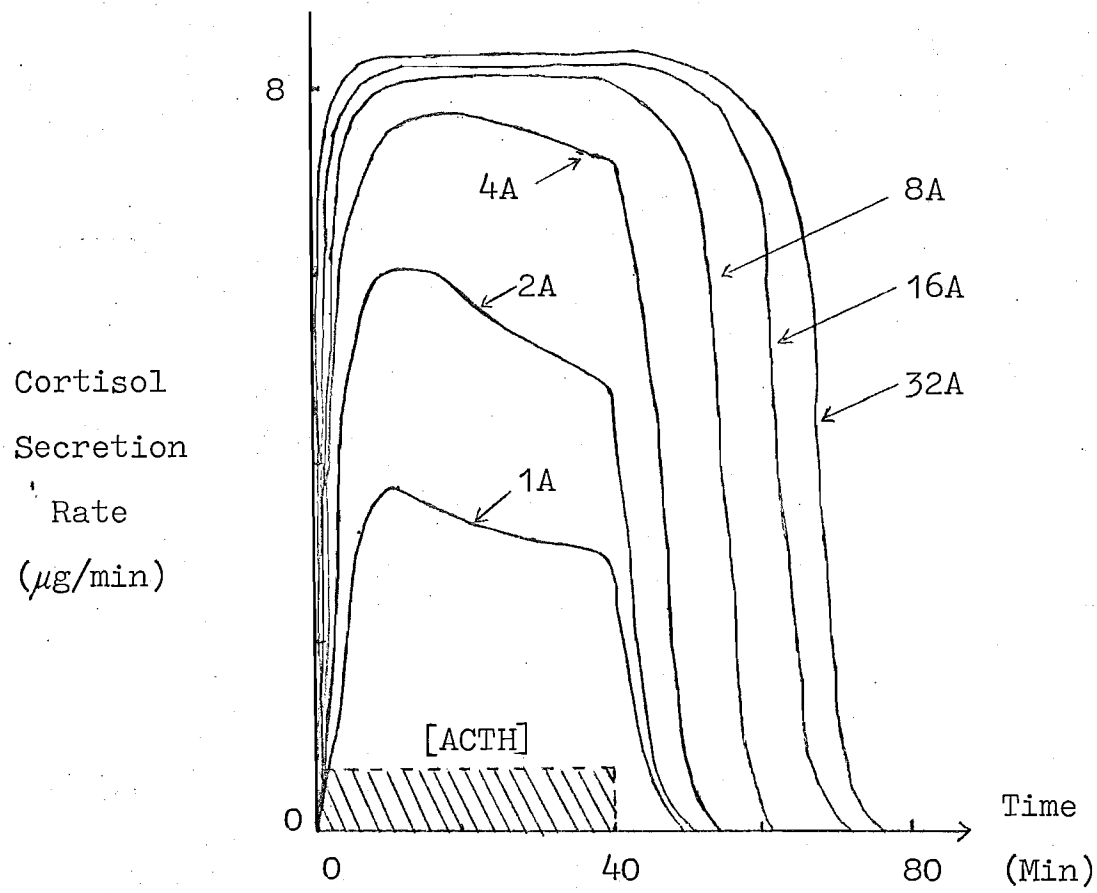


Figure 4.2 Response of storage model to a forty minute infusion of ACTH at varying levels. N.B. A = 1 arbitrary ACTH concentration unit.

Figure 4.2 shows that the response of the model is similar to that described in points 1 and 2 above, while figure 4.3 demonstrates the properties shown in points 4 and 5. The delay in the onset of the cortisol secretion (point 3) is not demonstrated by the model. This could be incorporated by delaying the action of ACTH, or alternatively by adding a transport delay between the tanks B and C.

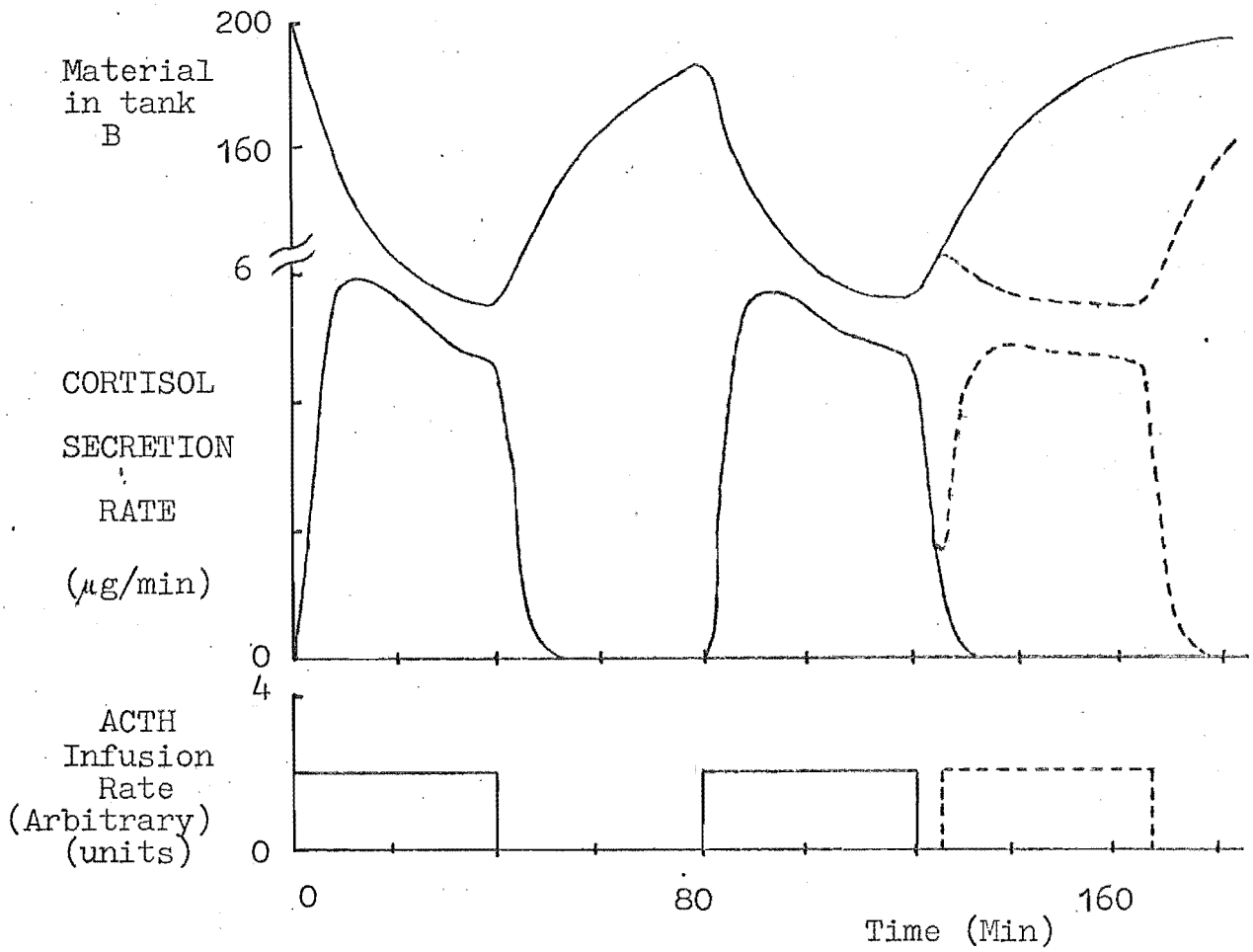


Figure 4.3 Response of storage model to repeated infusions of ACTH.

4.2 Identifying the Storage Model with the Adrenal Cortex

In the previous section (section 4.1) a model was developed which is capable of producing a response to changes of ACTH concentration similar in shape to that shown by the adrenal cortex. We now attempt to identify the processes incorporated in the model, with real processes in the gland.

The contents of each of the tanks in figure 4.1 could represent the amount of a precursor of cortisol which is present in the gland or in the bloodstream. The pipes between tanks could represent one of a number of processes - diffusion between regions of the gland, chemical transformations, or a combination of the two. The valve, by

which the flow from tank B to tank C is controlled, could represent a catalyst affecting a chemical reaction - the more catalyst present, the faster will the reaction proceed (in this situation the reverse reaction, which must always be present, is assumed to have negligible affect on the flow of material). The enlargement of tank C at the top could represent a saturating chemical reaction. That is, the rate of conversion of C to D is asymptotic to a fixed maximum level. Each of the processes incorporated in the model can therefore be described by biochemical processes.

We now return to the adrenal gland to investigate where such biochemical processes occur. Cholesterol, the major precursor of cortisol, is investigated first.

Cholesterol is found in blood at high concentrations ($\sim$1 mg/ml; Altman and Dittmer, 1961). From the blood, cholesterol moves into lipid droplets within the adrenal cells, where it is stored esterified to fatty acids (Gill, 1972). Before the 20α hydroxylation of cholesterol can occur, this substance must move into the cell mitochondria (Gill, 1972). It is within the mitochondria that ACTH acts, probably via the intermediate cyclic AMP (cf. section 2.1), to hydroxylate cholesterol and allow the synthesis of cortisol to proceed. Sayers et al. (1946) report that ACTH reduces the adrenal content of cholesterol, while Borkowski et al. (1972b) demonstrate that the amount of free adrenal cholesterol is only marginally affected by ACTH, and that cholesterol which has been esterified, is reduced by a factor of about 7 in the presence of this hormone.

As a result of the facts presented above we tentatively propose that esterified cholesterol is substance B in the model described above. Figure 4.4 shows the relationships

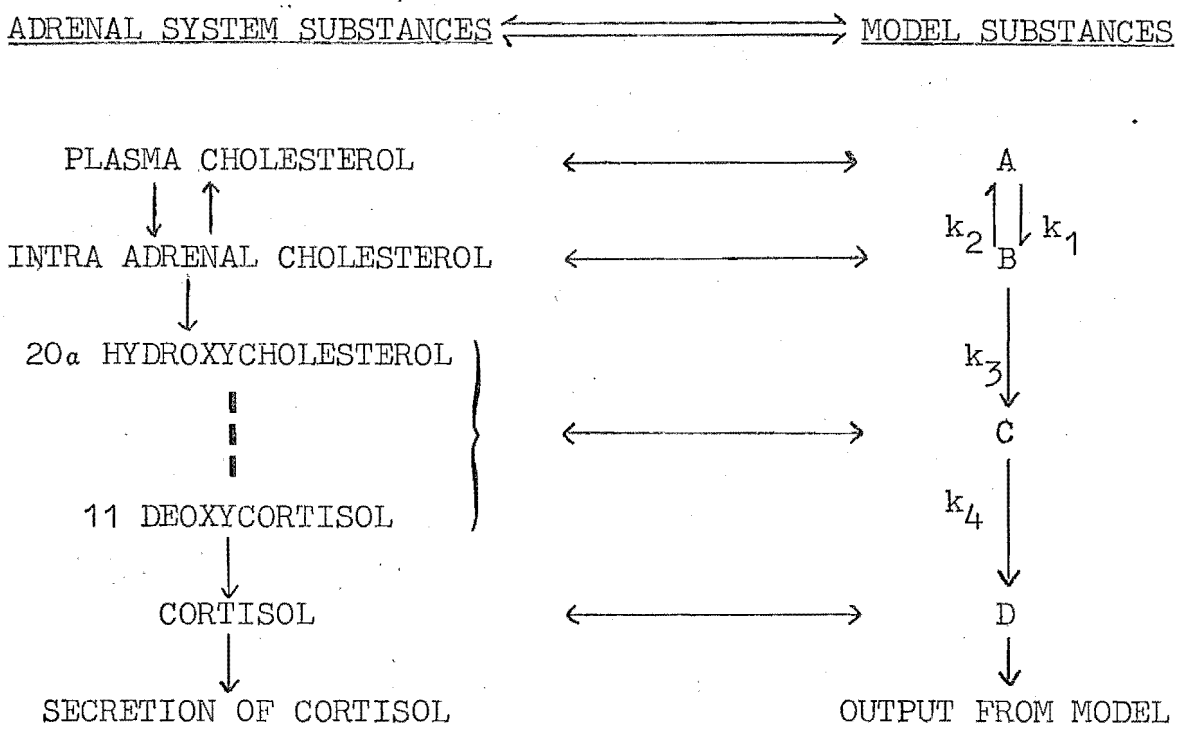


Figure 4.4. Postulate relationships between variables in the model of figure 4.1 and the chemical precursors of cortisol.

between substances A, B, C and D of the model, and the cortisol precursors. Note that substance C represents a number of cortisol precursors. Because many of the reactions in this pathway will occur very rapidly in the gland, describing a group of reactants and products as a single substance will have little effect on the result (this is justified in section 2.3).

Closer inspection of the data of Borkowski et al. (1972b), which is derived from studies on human subjects, shows that the amount of esterified cholesterol in a pair of adrenal

glands is 340 mg. This amount is sufficient to maintain adrenal steroid secretion for about seven days. Under intense ACTH stimulation the amount of esterified cholesterol is shown to drop to about 65 mg. Because this drop cannot be accounted for by cortisol secretion alone, we may deduce that not all of the cholesterol ester, which has been hydrolysed to form free cholesterol, is subsequently converted to steroid hormones, such as cortisol.

So that this difference between cholesterol ester hydrolysis and steroid hormone synthesis may be accounted for, the sizes of compartments A and B in the model must be increased, and the proportion of the material flowing from B that reaches C must be reduced.

The storage model developed in this section qualitatively accounts for the overshoot in cortisol secretion being caused by depletion of the amount of esterified cholesterol present in the gland. Quantitative verification of this is not possible until better data, describing the amounts of different forms of cholesterol present in the gland as functions of time, becomes available.

Whereas cholesterol has been strongly implicated as the stored substance in earlier discussion, this does not mean that it is the only one that satisfies the requirements of the model. There are many other substances actively involved in steroidogenesis, including precursors, cofactors and enzymes, some of which could be used equally well in place of cholesterol in the storage model. However, there is insufficient experimental support to justify investigating these substances in the model at this stage.

4.3 A Model of Berger's Cyclic AMP Proposal

Berger (1971) proposes that cyclic AMP, the intracellular mediator of the steroidogenic response, may initiate at least two patterns of adrenal steroid hormone synthesis, depending on the plasma concentration of ACTH. The proposal is based on evidence which shows cyclic AMP to stimulate . (Karaboyas and Koritz, 1965) and inhibit (Koritz et al., 1968) certain of the biochemical steps in the steroidogenic pathway. Berger's conclusions are summarized in figure 4.5.

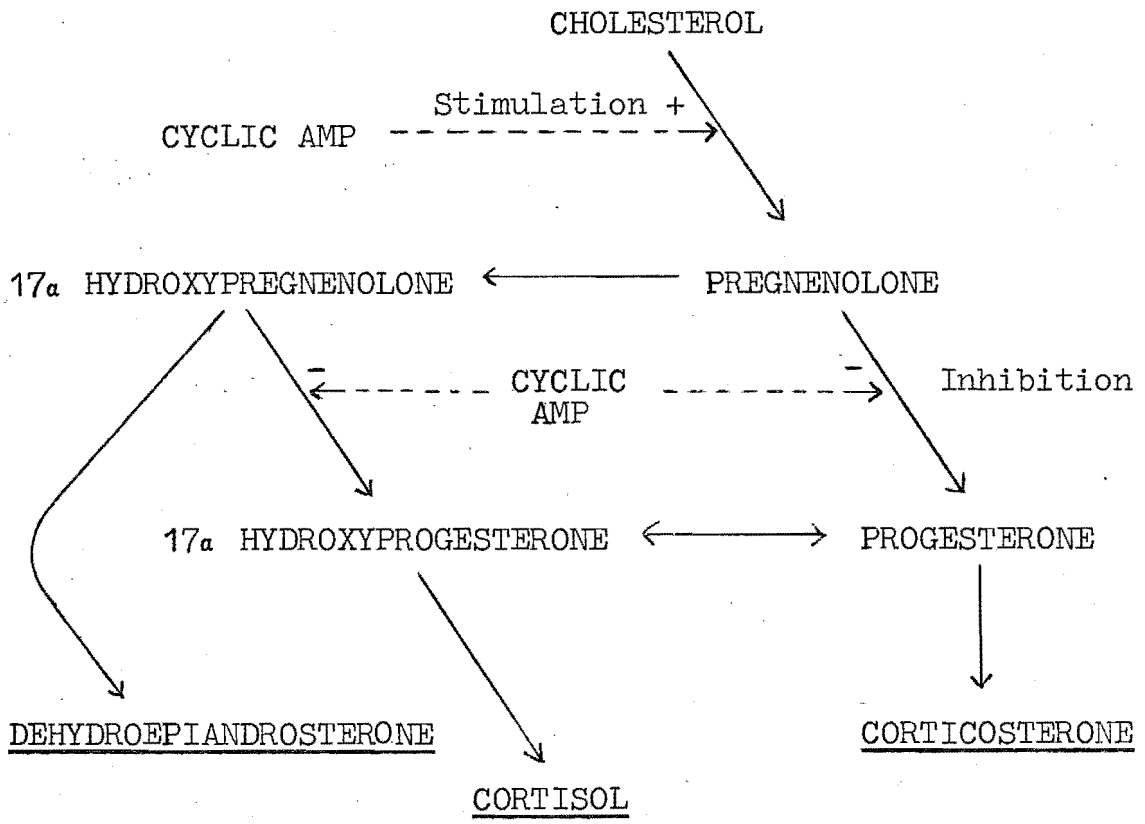


Figure 4.5 Summary of Berger's proposal for the action of cyclic AMP on steroidogenesis. From Berger (1971).

We enquire whether the proposal of Berger would explain the overshoot observed when the adrenal gland is stimulated by an infusion of ACTH.

By modelling the mechanism proposed by Berger (1971), it is found that the observed response of the adrenal cortex to ACTH may be duplicated, provided the inhibiting processes are dynamically slower than the stimulatory process.

Figure 4.5 is simplified by merging two of the biochemical paths into one. The resulting synthesis path is shown in figure 4.6.

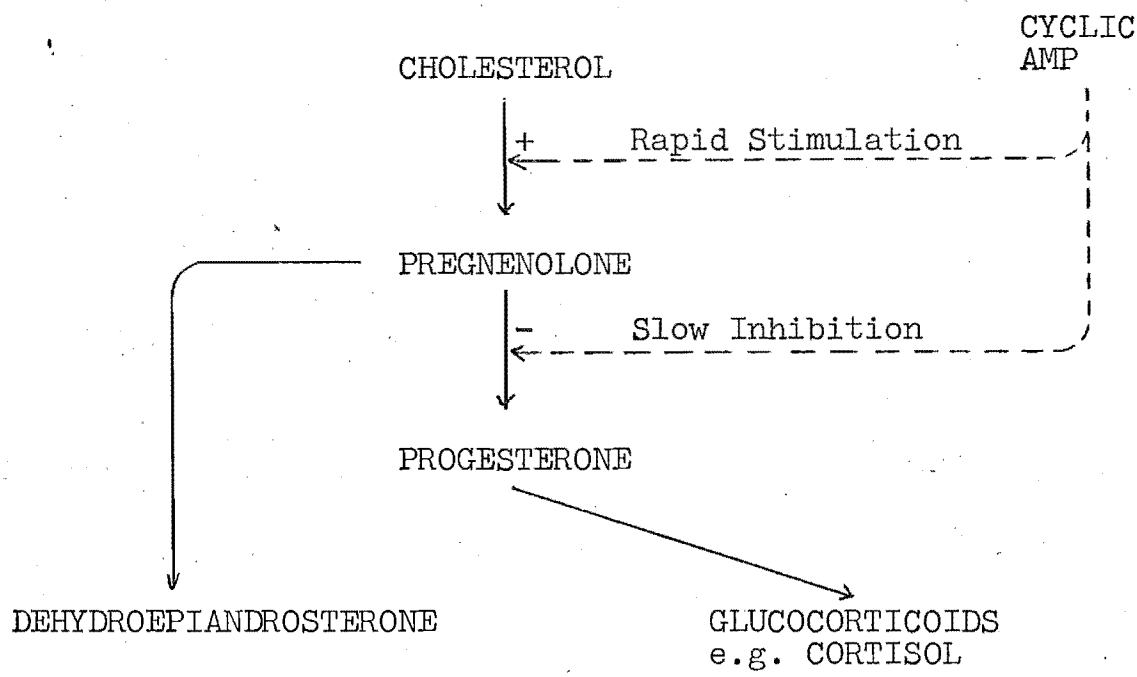


Fig. 4.6 Simplified version of Berger's cyclic AMP proposal, which was shown in figure 4.5.

In the model, the fast and slow actions of cyclic AMP are implemented by forming two compartments for this substance; the first affects only pregnenolone formation, while the second, which has a relatively long time constant, inhibits progesterone formation. The model is shown in compartment form in figure 4.7. The operation of the model is now described. Upon the introduction of ACTH to the adrenal, compartment CAMP (A) rapidly reaches equilibrium

and initiates steroid production. Compartment CAMP (B) fills slowly by comparison, and thus slowly decreases cortisol secretion by inhibiting the formation of progesterone. In this inhibitory process, another steroid, DHEA (dehydroepiandrosterone), is produced in greater quantities as the inhibition becomes more complete. By the process just described, the overshoot in the secretion rate of cortisol is formed.

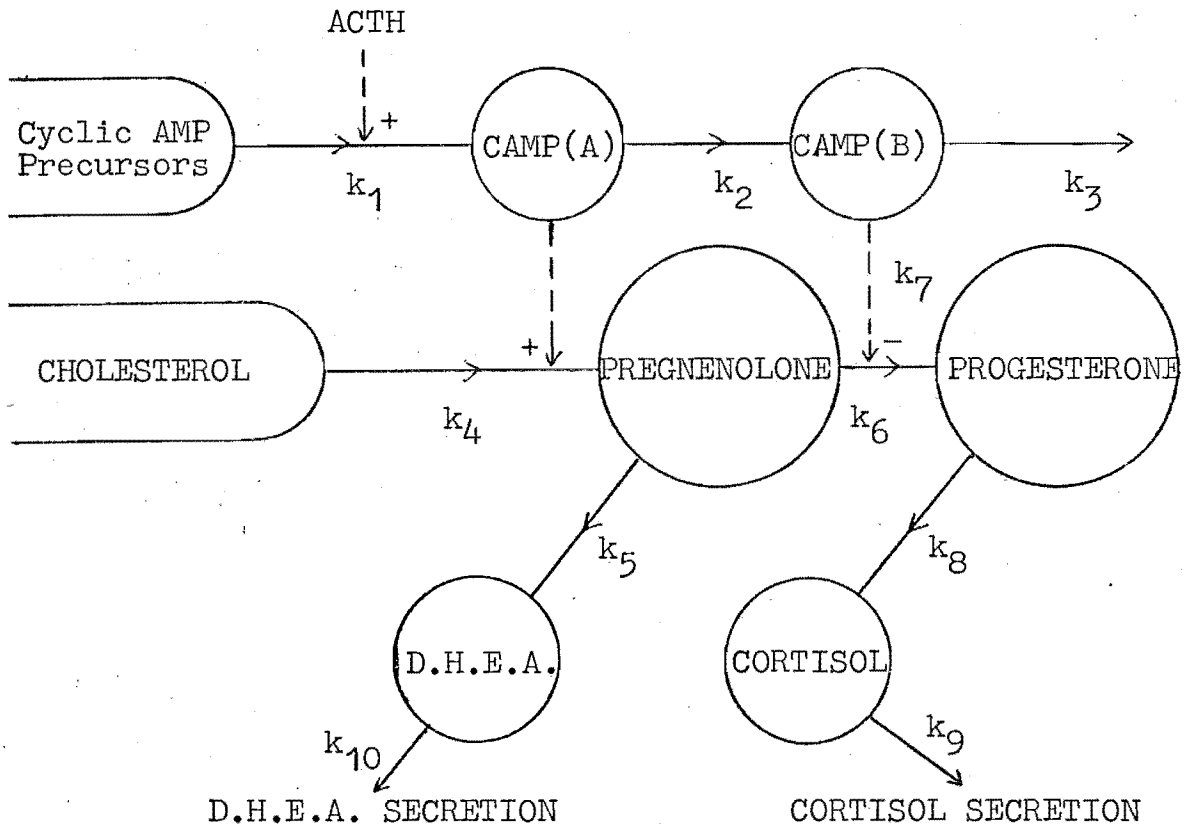


Figure 4.7 Model of steroidogenesis based on proposal put forward by Berger (1971).

The equations of the Berger model are derived from mass conservation on the compartments in figure 4.7.
These equations are

$$\dot{CAMP}\ (A) = k_1 \times ACTH - k_2\ CAMP\ (A)\ , \qquad (4.6)$$

$$\dot{CAMP}\ (B) = k_2 \times CAMP\ (A) - k_3 \times CAMP\ (B)\ , \qquad (4.7)$$

$$\dot{PREG} = CHOL \times CAMP\ (A) \times k_4 - PREG\ (k_5 + \frac{k_6}{1 + k_7 \times CAMP\ (B)})\ , \qquad (4.8)$$

$$\dot{PROG} = k_6 \times PREG\ /\ (1 + k_7 \times CAMP\ (B)) - k_8 \times PROG\ , \qquad (4.9)$$

$$\dot{GLUC} = k_8 \times PROG - k_9\ GLUC\ , \qquad (4.10)$$

$$\dot{DHEA} = k_5 \times PREG - k_{10} \times DHEA. \qquad (4.11)$$

The Berger model produces excessive overshoot at high ACTH concentrations. To eliminate this, it is necessary to add a saturating mechanism to one of the later chemical transformations. Thus, equations (4.9) and (4.10) become

$$\dot{PROG} = k_6 \times PREG\ /\ (1 + k_7 \times CAMP\ (B)) - k_8 \times PROG\ /\ (1 + k_{11} \times PROG)\ , \qquad (4.12)$$

and

$$\dot{GLUC} = k_8 \times PROG\ /\ (1 + k_{11} \times PROG) - k_9 \times GLUC\ , \qquad (4.13)$$

respectively.

The response of the Berger model, with the parameter values listed in table 4.8, are shown in figures 4.9 and 4.10.

Table 4.8 Values of Berger model parameters used in figures 4.9 and 4.10.

Parameter	Value	Parameter	Value	Parameter	Value
k_1	1	k_5	0.5	k_9	1
k_2	1	k_6	0.5	k_{10}	1
k_3	0.067	k_7	0.067	k_{11}	1
k_4	1	k_8	1	CHOL	1

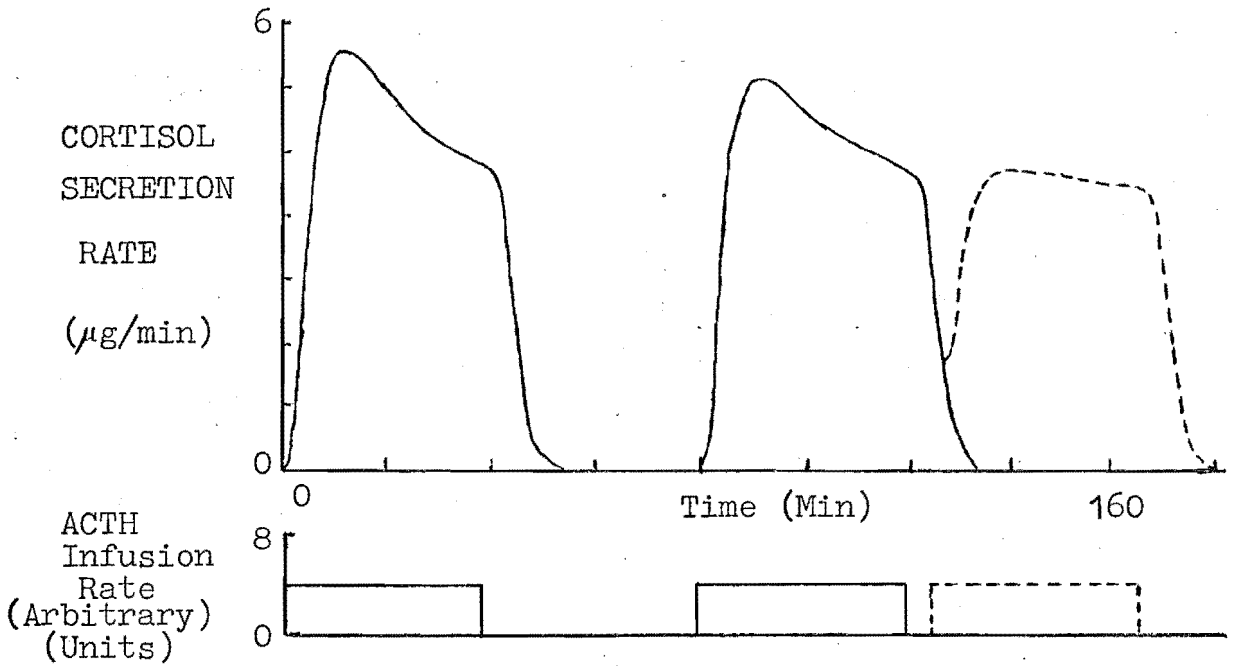


Figure 4.9 Response of Berger model to repeated infusions of ACTH.

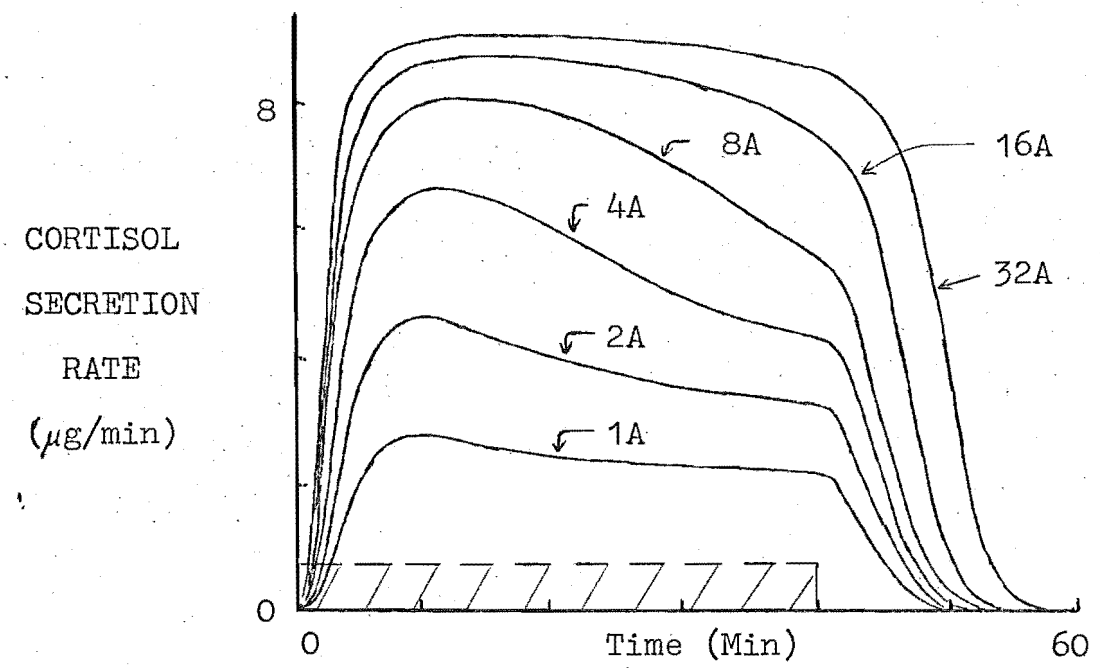


Figure 4.10 Response of Berger model to forty minute infusions of ACTH at varying levels. N.B. A = 1 arbitrary ACTH concentration unit.

4.4 Three Adrenal System Models?

We now have three models, each based on a completely different theory, and all modelling the one system. Undoubtedly, others are possible. Only one of these models, if indeed any of them, can provide a truthful account of reality. Which then falls into this category?

The Koritz-Hall model is partially discounted in section 4.0 because it contradicts experimental evidence that ACTH primarily controls the first cholesterol hydroxylation. This reaction is described by Berger (1971) and Gill (1972), as the rate-limiting step in the steroidogenic pathway. But in all three models (viz. the Koritz-Hall, Berger and cholesterol storage models), it was found necessary to incorporate a saturating chemical reaction in

the latter stages of the synthetic path. This implies that the first cholesterol hydroxylation may not be the only rate limiting reaction; at high ACTH concentrations, steroidogenesis is limited by other mechanisms in all three models. The three models are in agreement in that two independent processes in each model control the rate of steroid production. The first, which acts at low ACTH concentrations, differs amongst the three models, while the second, acting at high ACTH concentrations, is identical in all three. The importance of this last point is not that the saturating process is the same for each model, indeed it was designed that way, but that it is necessary to incorporate a saturation. This means that the saturation is an important part of the steroid synthesis, or alternatively, that none of the three models are models of reality.

To date no further models of the adrenal system have been developed, which do not require some form of saturation in the steroidogenetic pathway. From this we may tentatively infer that a saturating mechanism does exist.

The Berger model is based on postulates drawn from a number of in vitro experiments. It is possible that these experiments are not truly indicative of the in vivo system. The interactions which are observed to occur in incubations, may not be significant in the intact gland. For example, some of the chemical reactions will be either faster or slower in the in vitro preparations, thus making the interactions more dramatic, or, alternatively, making them

negligible. In this respect, further evidence is required to support the Berger hypothesis.

Returning to the operation of the Berger model, one method of testing it becomes obvious. In the period following ACTH administration, when cortisol secretion falls, the rate of production of DHEA increases as progesterone production is inhibited (cf. figure 4.7). From this we see that DHEA production will increase rapidly at the onset of ACTH administration and subsequently increase at a slower rate as the inhibition of progesterone synthesis becomes more complete. DHEA, or a metabolite of this substance must move out of the adrenal via the bloodstream, so the hypothesis of Berger may be tested by measuring adrenal secretions of DHEA or its metabolites. According to the model, the secretion rate of this substance will rise rapidly, following a step increase of ACTH concentration, in the initial few minutes, and then rise more slowly, reaching a peak in about forty minutes. On removal of the ACTH, DHEA secretion should drop rapidly to low levels, probably at a similar rate to that of cortisol.

The storage model, which is described here in terms of a storage pool containing esterified cholesterol, is based on less experimental evidence than the other two models discussed. This is not a sufficient reason to discount this model. Rather, it suggests that further research is needed to determine the movements of cholesterol in the gland during ACTH stimulation. The studies of Borkowski et al. (1972a; 1972b) go a long way in this direction, but do not provide sufficient information on the short term changes in

cholesterol concentrations in the various compartments to allow the model to be tested accurately.

The most significant point resulting from these modelling efforts is that the overshoot in the cortisol response cannot as yet be quantitatively explained in terms of experimental evidence. Each model is based on a little fact and a lot of theory. Consequently, much closer studies of the adrenal system, both in in vivo and in vitro situations, are needed.

CHAPTER 5

ADRENAL UPTAKE OF ACTH

Pearlmutter et al. (1971) suggest that ACTH must first bind to an ACTH specific binding protein on the adrenal cell membrane before it can stimulate cortisol synthesis. Urquhart and Li (1969) postulate that the ACTH molecule is inactivated in the process of stimulating the adrenal to produce cortisol, and tentatively propose the maximum rate of inactivation to be 100 μU (about 10 μg) of ACTH per minute per gram of adrenal tissue. Radioimmunoassay techniques for measuring physiological concentrations of ACTH (Donald, 1968) are now sufficiently sensitive to allow the above two proposals to be tested.

Experiments described in this chapter show that ACTH supplied to the artery of the adrenal gland does not all appear in the vein. It is shown that this is not due to leakage of water or ACTH, nor to delays in the transport of blood through the gland. ACTH must therefore be bound or otherwise destroyed in the adrenal.

The experiments described in this chapter were performed at Lincoln College by Dr E.A. Espiner of Princess Margaret Hospital, the ACTH measurement by Dr R.A. Donald of Princess Margaret Hospital, and the analysis of results by myself. The experimental procedures are described by Espiner et al. (1974).

5.1 Apparent Uptake of ACTH

During the course of one experiment, in which the concentrations of cortisol and ACTH in the adrenal venous blood of a ewe were monitored, it was found that cortisol was being secreted by the gland in the absence of significant levels of ACTH. This result is shown in figure 5.1. In the period from 70 to 100 minutes the data suggest that the

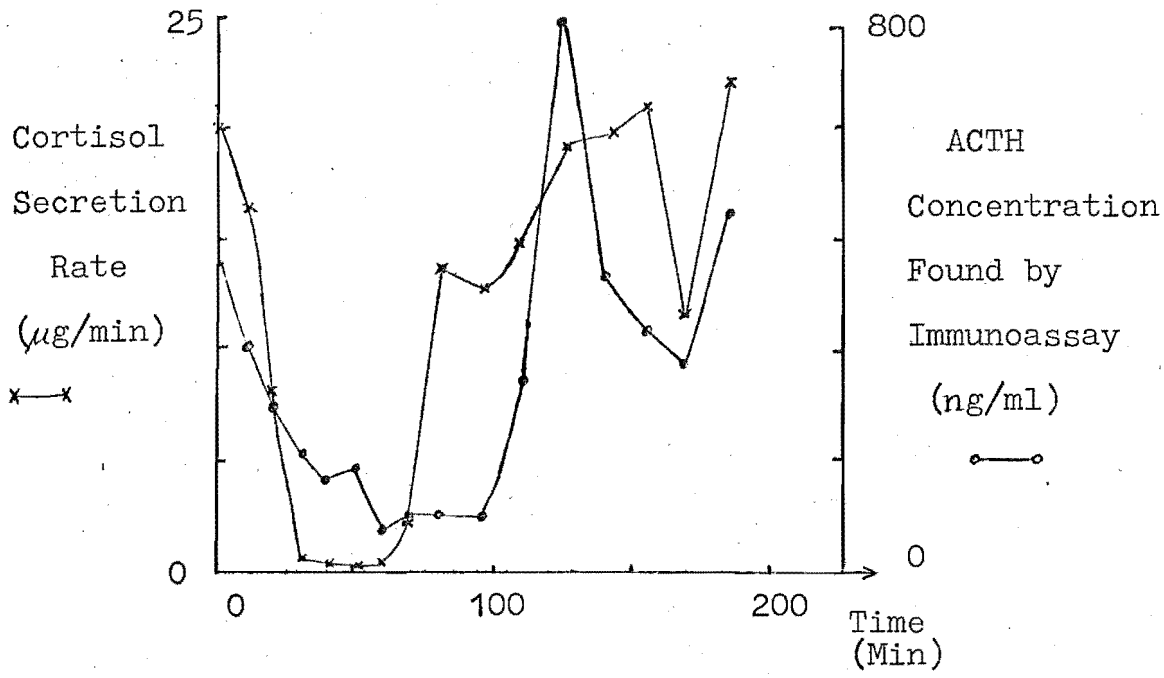


Figure 5.1 ACTH concentration and cortisol secretion rate measured in adrenal venous blood of a ewe. Note the high cortisol secretion and low ACTH concentration in the period 70 to 100 minutes. This is unexpected because ACTH stimulates cortisol production.

adrenal might be inactivating or binding ACTH. However, because only two data points show the anomaly, the result is inconclusive.

As a consequence of the findings from this experiment, further experiments were undertaken by the Lincoln Group to observe the fate of ACTH in its passage through the adrenal gland.

5.2 Measurement of Arterio-Venous ACTH Concentration Differences

Five merino ewes with adrenal glands autotransplanted to a jugular carotid loop (cf. section 3.3) were used by the Lincoln group in experiments to measure ACTH concentrations in adrenal venous and arterial blood. A sixth animal with a similar jugular-carotid loop but without a transplanted adrenal gland was used as a control for the experiment.

Dexamethasone was injected into each animal at least four hours before the experiment, to stop the production of ACTH in the pituitary of the sheep.

ACTH was infused into the blood of each animal at a point remote from the gland. An infusion rate of 0.5 mg/min was maintained for a period of 130 minutes. Blood samples taken from the adrenal artery and vein at intervals of between five and ten minutes were later analysed for ACTH by radioimmunoassay. Blood from the adrenal vein was also analysed for cortisol and the rate of secretion of this hormone calculated. The mean ACTH concentrations and cortisol secretion rates from the five animals are shown in figure 5.2.

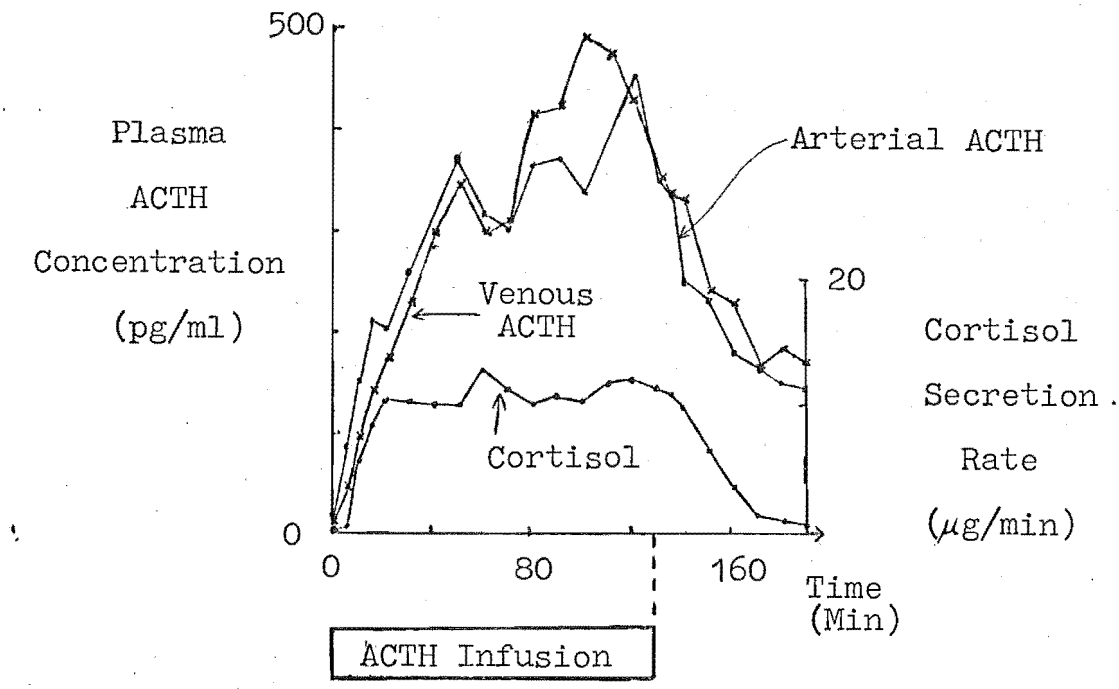


Figure 5.2 Adrenal arterial and venous concentrations of ACTH, and cortisol secretion rate during a constant infusion of ACTH. The mean values from five experiments for each quantity are shown in the figure.

Although arterial ACTH concentrations do exceed venous concentrations in figure 5.2, the difference is not as great as required to explain the results shown in figure 5.1. We enquire whether the arterio-venous concentration difference shown in the first forty minutes is statistically significant, and whether this difference can be explained by transport delays through the gland, or by loss or gain of water or ACTH through the extravascular fluid.

So that the effect of variations in infusion rates, and animal size in each experiment, may be minimized in the analysis of the results, it is necessary to normalize the data from individual experiments. The data values from each

of the five experiments are normalized by multiplying them by the plasma volume of the animal being analysed, and dividing by the rate of ACTH infusion. The normalization is thus

$$ACTH_{NORM} = \frac{[ACTH]_{ACTUAL} \times VOL_{PLASMA}}{\text{Infusion Rate}} \qquad (5.1)$$

Blood was collected at the adrenal artery at a constant rate over a period of one half minute, for each sample. The sample time is taken to be the centre point of the sampling period. With the venous samples a further complication arises. These samples are extracted by siphoning from the adrenal vein for a fixed period of time - in this case one minute. Because the volume of the siphon is 9 ml and the flow rate of blood in the siphon between 6 and 25 ml/min, the collection of the sample is delayed up to $1\frac{1}{2}$ minutes in transit through the siphon tube. The true time of the venous sample (T_{VTRUE}) is therefore taken as

$$T_{VTRUE} = T_{VMEAS} - 9 \text{ ml} / \text{Blood flow rate} , \qquad (5.2)$$

where T_{VMEAS} is the time at the mid point of the sampling period.

Because the data for arterial and venous samples occur at different times, the difference between arterial and venous ACTH concentrations cannot be found directly. Instead, linear regression lines are fitted to the pooled arterial and venous data, over three time periods (viz. 0 to 20, 20 to 40, and 40 to 60 minutes), and the arterio-venous differences calculated by adding the perpendicular distances

from the arterial and venous points to the regression line (see figure 5.3). Using a students t statistical test, the

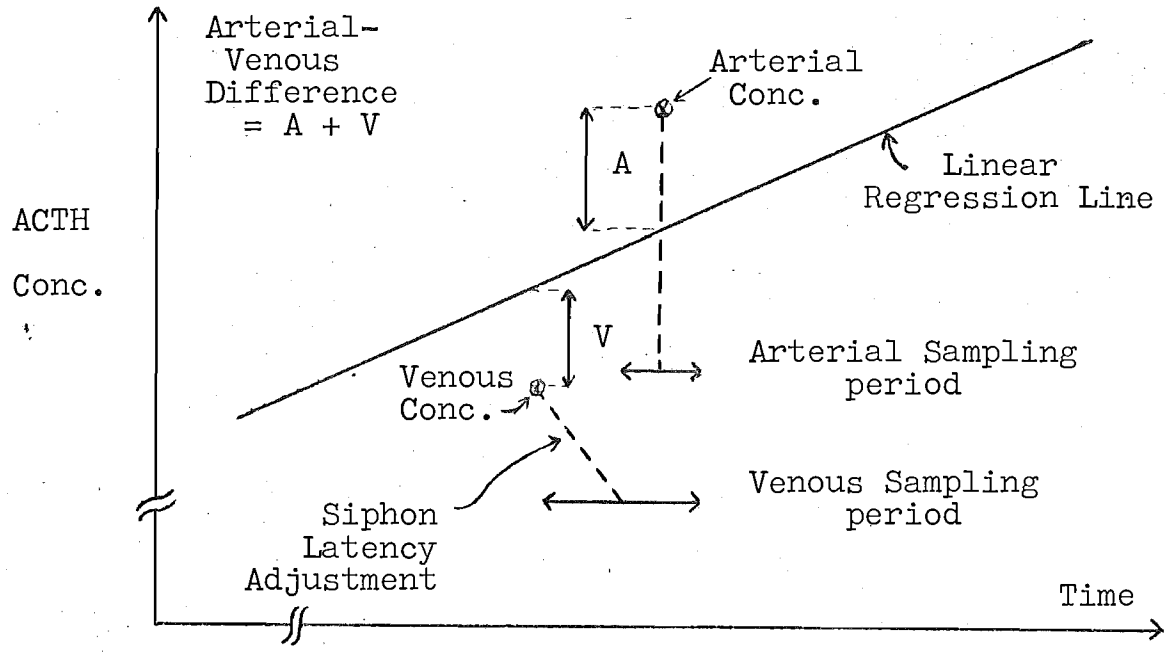


Figure 5.3 Method of correcting for siphon tube latency and estimating concentration difference between arterial and venous samples using linear regression lines.

difference between arterial and venous concentrations is found to be highly significant over the time periods 0 to 20 and 20 to 40 minutes ($p < 0.001$, and $p < 0.02$ respectively), and not significant in the 40 to 60 minute period ($p > 0.1$). In figure 5.4, the data from the five transplant animals are shown with estimates of their standard errors. Also shown is the data from the control animal, which was subjected to the same experiment. The difference between arterial and venous concentrations is not maintained in the control animal, and in fact venous concentrations exceed arterial concentrations for some of the time.

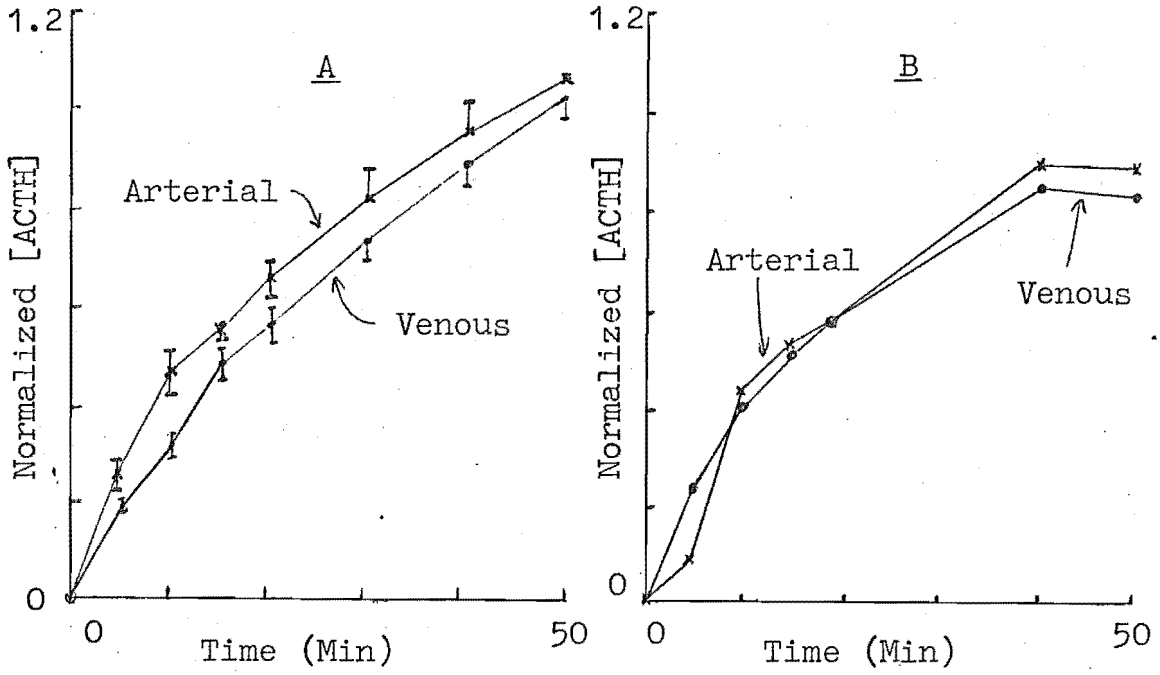


Figure 5.4 Normalized ACTH concentrations in arterial and venous blood as functions of time, A - means and standard errors from five transplant animals, B - values from control animal without a transplanted gland.

5.3 Measurement of ACTH Accumulation or Loss

The data shown in figure 5.4 gives an indication of possible rates of uptake or loss of ACTH, in its passage through the gland. The total amount of ACTH disappearing may be found by integrating the data of figure 5.4. Rather than integrating figure 5.4 directly, the data from each experiment is integrated separately and a mean of the resulting values is taken. In this study no attempt is made to normalize the data from each animal. Figures for the most

appropriate normalizing factor - namely adrenal weight - are not available for these animals.

The accumulated inflow and outflow of ACTH is found by integrating separately the product of blood flow rate, and the arterial and venous ACTH concentrations respectively. Trapezoidal integration is used. Thus, (Thomas, 1964, p.210)

$$\int_{T_n}^{T_{n-1}} \text{ACTH}\, dt = ([\text{ACTH}](T_n) \times BF(T_n) + [\text{ACTH}](T_{n+1}) \times BF(T_{n+1})) \times \frac{T_{n+1} - T_n}{2}, \qquad (5.3)$$

where $[\text{ACTH}](T_n)$ is the concentration of ACTH at time T_n, and $BF(T_n)$ is the blood flow rate measured while taking the venous sample. We make the assumption here that arterial and venous blood flow rates are equal. In other words there is no net loss or gain of water in the passage of blood through the gland. This assumption is justified in section 5.4. The accumulation of ACTH from sample to sample is found using

$$\int_0^{T_{n+1}} \text{ACTH}\, dt = \int_0^{T_n} \text{ACTH}\, dt + \int_{T_n}^{T_{n+1}} \text{ACTH}\, dt. \qquad (5.4)$$

The resulting integrals of arterial and venous ACTH for each animal are then linearly interpolated to give estimates of accumulated ACTH into and out of the gland at fixed times. The apparent uptake of ACTH is found by subracting outflow from inflow for each animal, at each fixed time. Means and standard errors of the apparent uptake are shown in figure 5.5.

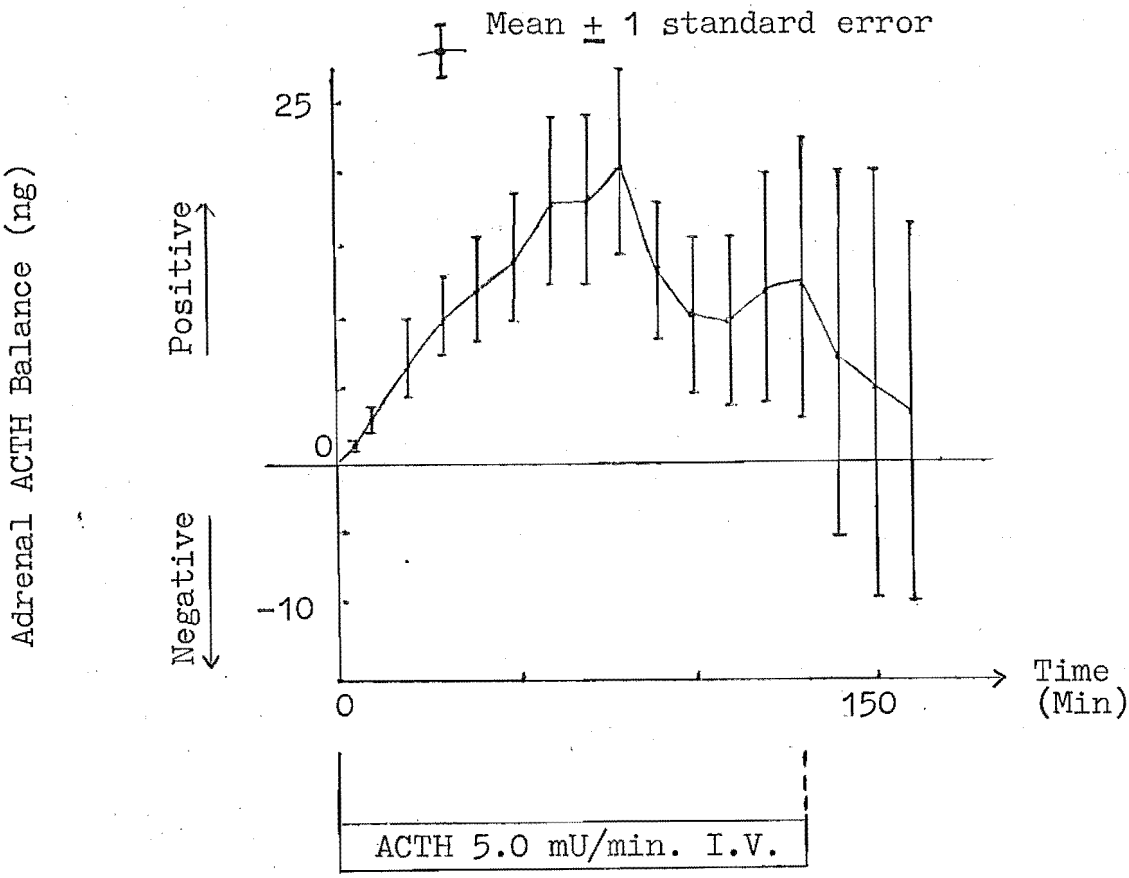


Figure 5.5 Mean and standard errors of the apparent uptake of ACTH by the gland during an infusion of ACTH. Figure shows mean data from five experiments on different animals.

The apparent uptake rises steadily, reaching a peak value of 20 ng at 80 minutes. Throughout this first 80 minutes the mean uptake is significantly greater than zero ($p < 0.005$). In the period 90 to 160 minutes in figure 5.5 the apparent uptake decreases with time. However, during this period the standard errors increase, which shows that between animal variation is greater here. This variation is expected because the raw data is not normalized.

5.4 Investigation of ACTH Accumulation or Loss

The apparent difference between arterial and venous concentrations may be caused by one of four processes:

1. ACTH is taken up by, or is inactivated within, the adrenal gland.
2. the movement of ACTH through the gland is slow.
3. water is gained from the extravascular fluid thus lowering the venous concentrations of ACTH.
4. ACTH is selectively lost to the extravascular fluid.

In this section postulates 2 and 3 are investigated by studying the results of measurements of, firstly the transport of an inert substance through the gland, and secondly the movement of ACTH infused directly into the adrenal artery.

Postulate 4 can be discounted immediately. Any movement of ACTH from the vascular to the extravascular fluid must be associated with a movement of water. Because water can move more easily than ACTH, less ACTH will move from the vascular system than will water, which will tend to increase the ACTH concentration in the adrenal vein. As the venous concentration is observed to decrease, this possibility is excluded.

The experiments described in section 5.2 were repeated by the Lincoln group using albumin, an inert plasma protein, which had previously been labelled with radioactive iodine. The labelled albumin is hereinafter called RISA (Radio-Iodinated Serum Albumin). The data from these experiments, normalized, corrected for sampling latency and averaged in

the same way as is described in section 5.2, are shown in figure 5.6.

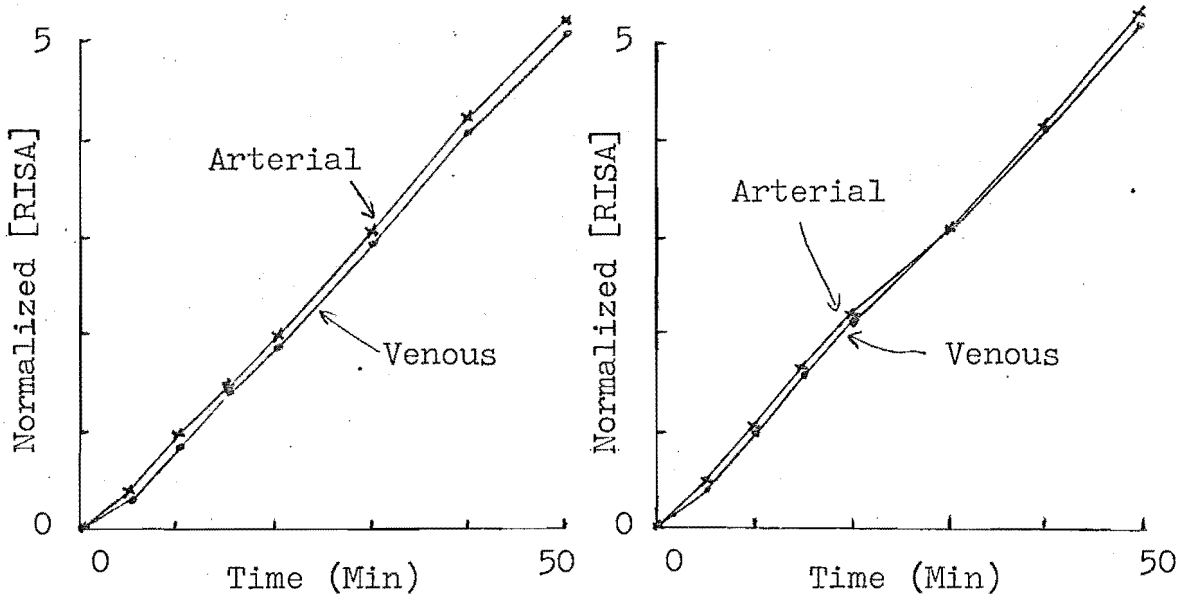


Figure 5.6 Normalized RISA concentrations in arterial and venous blood as functions of time. A - means from five transplant animals. B - values from the control animal.

The data from the RISA infusion experiments show that transport delays through the gland are of the order of one minute in duration, and do not exceed one and a half minutes at any stage (cf. figure 5.6). Inspection of figure 5.4 shows that arterial and venous curves are separated by at least five minutes. The possibility that the difference in ACTH concentrations across the gland is caused by transport processes alone, is therefore excluded.

The RISA infusion experiments described above suggest that the arterial and venous blood flow rates are equal. The large size of the RISA molecule (MW $\simeq$ 69,000) ensures

that it is constrained to move within the vascular system. Any gain or loss of fluid at the adrenal gland would be reflected as a proportionate decrease or increase respectively of the concentration of RISA. This is not apparent in figure 5.6, and we must therefore conclude that changes to the blood flow rate between the adrenal artery and adrenal vein are insignificant. These results therefore, discount postulate 3 above.

The values of plasma volume used in equation (5.1) are derived from the results of the RISA infusion experiments using

$$VOL_{PLASMA} = \frac{\frac{1}{2}\left([RISA]_{Art}\ (T_n) + [RISA]_{Ven}\ (T_n)\right)}{[RISA]_{Infusate} \times \text{Infusion rate} \times T_n}\ , \quad (5.5)$$

where T_n is the length of the infusion period. Equation (5.5) estimates the volume of distribution of albumin, which is probably significantly larger than that of ACTH because the molecular weights of RISA and ACTH are approximately 69,000 and 4,500 respectively. The difference between the volumes of distribution of the two substances is discussed later (cf. section 6.6).

The experiments described above allow estimation of the time delays for RISA transport through the gland. Further evidence is required to show that these delays also apply to the ACTH molecule, which is considerably smaller. To study this, the Lincoln group infused ACTH into the adrenal artery of each of the transplant animals at a constant rate of 30 μg/min for fifteen minutes. Adrenal venous blood was siphoned off into collection vessels, each taking a sample

for two minutes. The samples were then analysed for ACTH as before. Figure 5.7 shows the accumulated ACTH in the venous effluent as a function of time, averaged over the five animals.

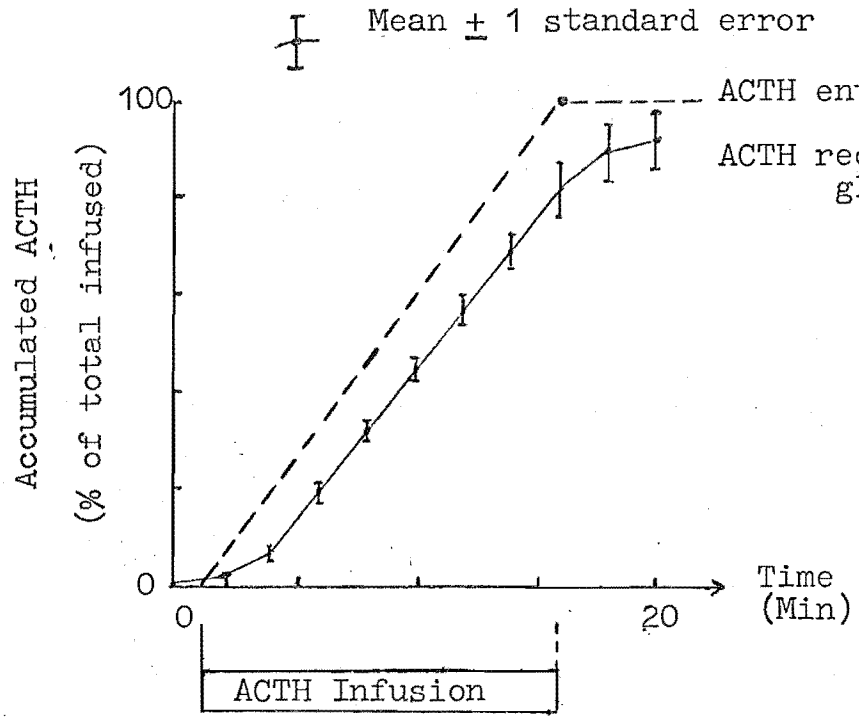


Figure 5.7 ACTH recovered at the adrenal vein following an infusion into the adrenal artery. Mean and standard errors from five experiments are shown.

Also shown is the theoretical amount of ACTH entering the gland through the adrenal artery. From the figure it is apparent that nearly 95% of the ACTH entering the gland is recovered and that at the end of the measurement period the amount recovered is still increasing. The maximum delay in transport of ACTH does not exceed three minutes at any stage. It should be noted here that the rate that ACTH enters the gland in these experiments (viz. 20.0 to 50.0 μg per minute) is significantly greater than in the experiments discussed in section 5.2 (viz. 5 to 20 ng per minute). At this

concentration, any uptake of ACTH would be negligible by comparison.

Both of the experiments described in this section show that the arterio-venous concentration difference observed in section 5.2 cannot be explained by time delays through the gland alone, thus invalidating postulate 2 above. These experiments also provide further evidence against postulate 4, and invalidate postulate 3.

5.5 Interpretation of Uptake Experiments

While the evidence presented here shows that an arterio-venous difference in ACTH concentration does occur, the data does not allow the cause of this difference to be established. The observed difference can be the result of two mechanisms -

- adrenal tissue may be binding the ACTH molecules.
- adrenal tissue may be changing the ACTH molecules in such a way that they are not measured by immunoassay.

Either of these processes might be associated with the known action of ACTH, i.e. stimulation of cortisol synthesis.

While the radioimmunoassay technique, used for the measurement of ACTH concentration, is sufficiently sensitive for this analysis, there is evidence that another form of ACTH, which is itself not biologically active, is also measured by the assay. The sensitivity of the assay to the different forms of ACTH is discussed in more detail in chapter 6. There is a possibility that the ACTH measurements described in this chapter are contaminated by the inactive forms of the molecule. This could mean that the venous samples may contain a proportion of inactive ACTH. If this

were so the level of adrenal inactivation of ACTH might be larger than has been shown here.

Before further interpretation of the level of adrenal uptake or inactivation of the ACTH molecule may be made, more specific ACTH assays must be used in studies similar to those cited in this chapter.

CHAPTER 6

ACTH FRAGMENTATION MODEL

In the previous chapter a group of experiments was discussed, which showed possible disagreement between the ACTH concentration (as measured by radioimmunoassay) present at the adrenal gland, and the rate of secretion of cortisol by the glands. It has been proposed that this anomaly is caused by the measurement technique detecting fragments of the ACTH molecule, which do not have full biological activity.

In this chapter, a number of theories to account for the experimental results are investigated. It is shown that only those theories which reduce to nonlinear models, or models containing two or more compartments of ACTH or its derivatives, are capable of reproducing the results. A two compartment linear model, based on the postulate that ACTH fragments, which are not themselves biologically active, are being detected by immunoassay, is developed, and its parameters estimated by least squares parameter estimation. The concentrations of biologically active ACTH estimated from this model are consistent with the rate of secretion of cortisol by the adrenal gland. The model is further tested against data from the literature, in which both biological and immunological determinations of ACTH concentration are implemented. The results from fitting the model to these published experiments are inconclusive because of the few data points available for analysis.

The two compartment model is explained either by the fragmentation theory described above or by a second theory postulating inactivation of the ACTH molecule rather than its complete destruction.

6.1 Difference between Bioassay and Immunoassay Results.

A typical result of one of the experiments discussed in chapter 5 is shown in figure 6.1. When the infusion of

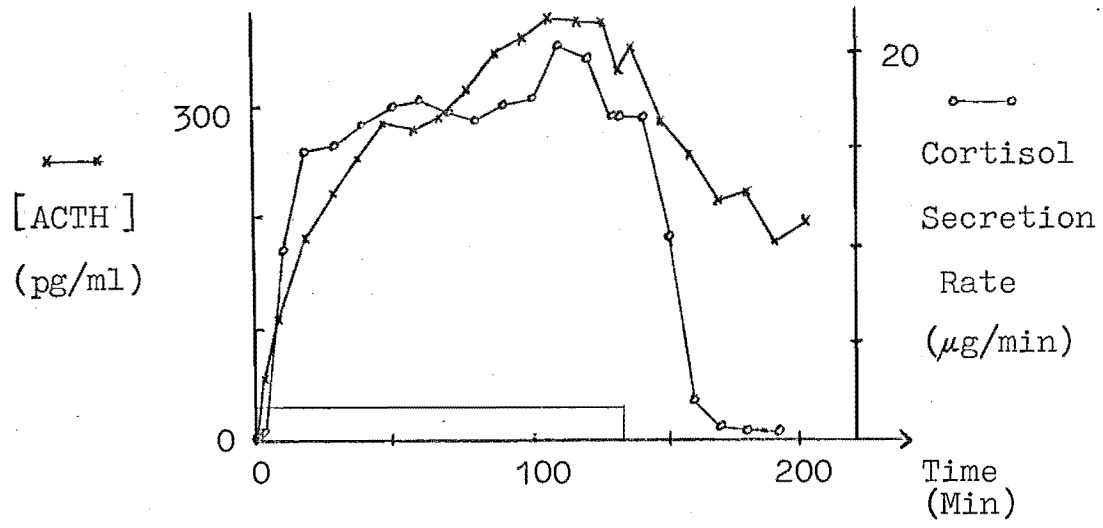


Figure 6.1 Measured ACTH and cortisol secretion rate during an infusion of ACTH into systemic blood of sheep.

ACTH is stopped, the secretion of cortisol drops rapidly, while ACTH concentration falls slowly by comparison. This is unexpected because ACTH stimulates the production of cortisol. This difference between measurements of bioactive and immunoactive ACTH has been reported before (Matsuyama et al., 1972; Murphy et al., 1969; Bessar et al., 1971). Bessar et al. (1971) propose that immunoassay measures not only ACTH but also fragments of the ACTH molecule which do not affect cortisol production. It is these fragments

that remain at a high concentration after cortisol secretion has fallen.

The experimental results cannot be explained by a single compartment breakdown of ACTH, because this would require the rate of decay of cortisol secretion to be as slow, if not slower than that of ACTH concentration. This result is not shown by experiment (cf. figure 6.1). Thus, the breakdown process must involve at least two forms of ACTH or its derivatives, and consequently, the model describing the breakdown must incorporate at least two compartments for ACTH.

6.2 Bessar's Theory of ACTH Fragmentation

The structure of, and the position of biological and immunological activity within, the ACTH molecule are important to Bessar's fragmentation theory. A brief account of these features of the ACTH molecule is now given.

ACTH is a chain like molecule with thirty nine links, each of which is derived from one of the twenty common amino acids. The links of the molecular chain are called amino acid residues or peptides; the whole molecule is therefore called a polypeptide. Fifteen different amino acid residues are found in the ACTH molecule. The complete structure of the ACTH molecule is given by Frieden and Lipner (1971).

Of the thirty nine peptides comprising the ACTH molecule, only the first twenty four show evidence of biological activity. Seelig and Sayers (1973) report that a derivative of ACTH containing only the first twenty four peptides (corticotrophin-(1-24)-tetracosapeptide) is a more potent

stimulator of steroid production than the entire ACTH molecule. Removal of the first four, or last eight peptides within the first twenty four (corticotrophin-(5-24)-eicosapeptide, and corticotrophin-(1-16)-hexadecapeptide amide, respectively) reduce the biological activity by more than a thousand fold (Seelig and Sayers, 1973).

Bessar's fragmentation theory is shown diagrammatically in figure 6.2. The ACTH molecule can break at any of

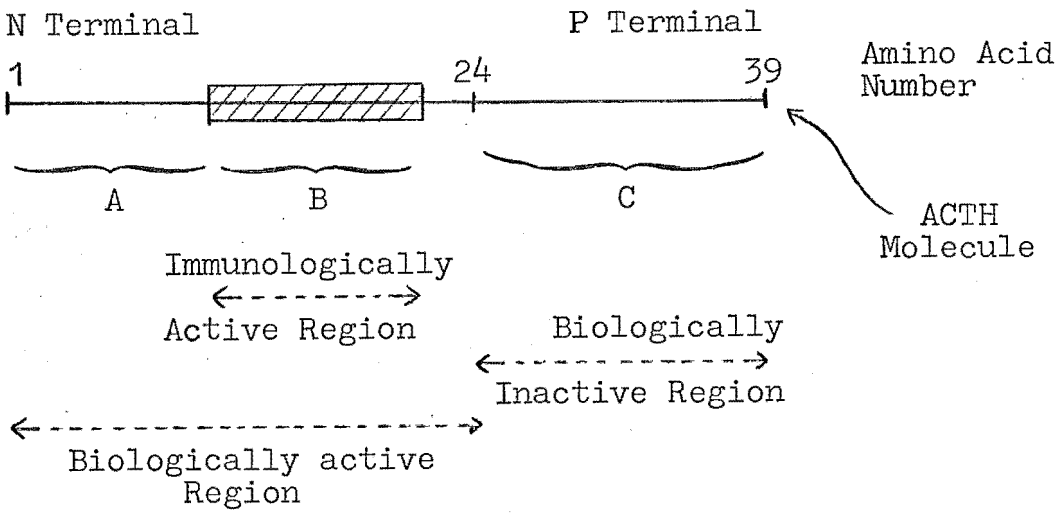


Figure 6.2 shows the possible configuration of the ACTH molecule in relation to its measurement and breakdown. Breakage at A destroys only biological activity. Breakage at B destroys both biological and immunological activity. Breakdown at C does not affect biological or immunological activity.

several points within the regions A, B or C in figure 6.2. At A and C immunological activity is preserved, but biological activity is preserved only if the break is in the region C. Thus, if a break occurs in the region A the

remaining fragment will retain immunological activity, and will be detected by the immunoassay but will be biologically inactive. This theory would explain the results shown in figure 6.1.

6.3 The Fragmentation Theory in Model Form

The fragmentation theory relies on two forms of the ACTH molecule being distributed in the plasma system; the first, the complete ACTH molecule which retains full biological and immunological activity, and the second, a fragment of the molecule, which is biologically inactive but immunologically active. Figure 6.3 shows the fragmentation model using the compartment notation described in section 2.1. The biologically active ACTH (A in figure 6.3) decomposes by two mechanisms, one of which destroys biological activity but does not affect immunological activity (path k_2 in figure 6.3), and another which destroys both biological and immunological activity (path k_1 in figure 6.3). The biologically inactive form of ACTH (F in figure 6.3) is further broken down to a form which retains neither immunological nor biological activity (path k_3 in figure 6.3). Immunoassay detects a weighted sum of the two compartment concentrations [A] and [F] , the weighting factors (α and β) representing the sensitivity of the immunoassay to the two moieties. Note that, as the assay is calibrated against pure ACTH, the factor α is unity (i.e. $\alpha = 1$).

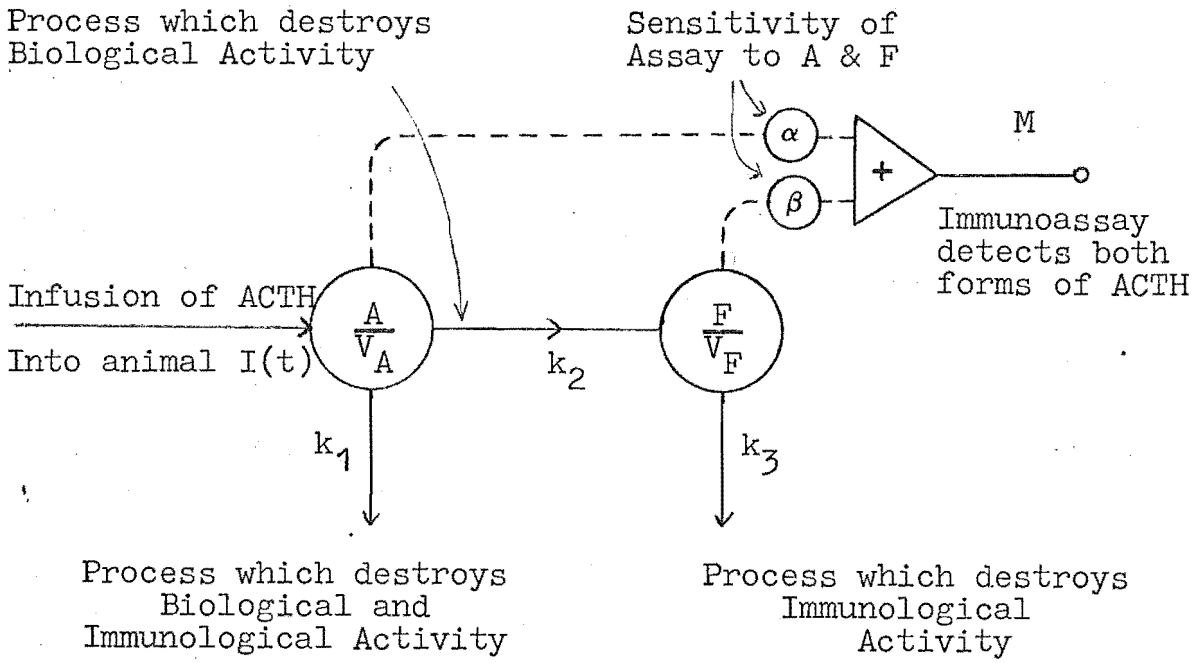


Figure 6.3 Compartment model of ACTH breakdown and measurements. A - amount of biologically active ACTH in the animal (pg ACTH), V_A - volume of distribution of biologically active ACTH (ml), F - amount of immunological ACTH that is not biologically active, V_F - volume of distribtuion of substance F (ml).

Using the law of conservation of mass, the equations of the model are now derived. For compartment A

$$\dot{[A]} . V_A = I(t) - (k_1 + k_2) . [A] . V_A , \qquad (6.1)$$

and for compartment B

$$\dot{[F]} . V_F = k_2 . [A] . V_A - k_3 . [F] . V_F . \qquad (6.2)$$

The measured substance M is related to A and F by

$$[M] = [A] + \beta . [F] \tag{6.3}$$

In the experiments under consideration, the forcing function I(t) is a constant infusion of J μg/min of ACTH from time t = 0 to time t = T. That is

$$I(t) = J . (u(t) - u(t - T)) , \tag{6.4}$$

where u(t) is the unit step function defined as

$$\begin{aligned} u(t) &= 0 \; ; \; t < 0 \\ &= 1 \; ; \; t > 0 \end{aligned} \tag{6.5}$$

Prior to starting the experiment both compartments are empty, i.e.

$$[A]_0 = [F]_0 = 0 . \tag{6.6}$$

The solution to equations (6.1), (6.2), (6.3) and (6.4) for the substance M is

$$[M](t) = \frac{J}{V_A}\left(\frac{k_3 + \beta k_2 \, V_A/V_F}{(k_1 + k_2) \, k_3} (1 - u(t - T)) - \right.$$

$$\frac{k_1 + k_2 - k_3 - \beta k_2 \, V_A/V_F}{(k_1 + k_2)(k_1 + k_2 - k_3)} \left(e^{-(k_1+k_2)t} - u(t-T) e^{-(k_1+k_2)(t-T)} \right) -$$

$$\left. \frac{\beta k_2 \, V_A/V_F}{k_3 \, (k_1 + k_2 - k_3)} \left(e^{-k_3 t} - u(t - T) \, e^{-k_3 (t - T)} \right) \right) . \tag{6.7}$$

Equation (6.7), during the period of the infusion (0 < t < T), reduces to

$$[M]\ (t) \;=\; \frac{J}{V_A}\left(\frac{k_3 + \beta k_2\ V_A/V_F}{(k_1 + k_2)\ k_3} - \right.$$

$$\frac{k_1 + k_2 - k_3 - \beta k_2\ V_A/V_F}{(k_1 + k_2)(k_1 + k_2 - k_3)}\ e^{-(k_1 + k_2)t} -$$

$$\left. \frac{\beta k_2\ V_A/V_F}{k_3\ (k_1 + k_2 - k_3)}\ e^{-k_3 t} \right) , \qquad (6.8)$$

which has the form

$$[M](t) \;=\; J\ (B_0 + B_1\ e^{-b_1 t} + B_2\ e^{-b_2 t}) , \qquad (6.9)$$

where

$$B_0 + B_1 + B_2 \;=\; 0 . \qquad (6.10)$$

Because only four independent quantities may be estimated from the experimental data (viz. B_1, B_2, b_1, and b_2), there are insufficient quantities to determine the values of all six of the parameters k_1, k_2, k_3, V_A, V_F and β. However, by suitably grouping these parameters, it is possible to reduce the number of model parameters to four. Let

$$k_{12} \;=\; k_1 + k_2 \;\; ; \;\; k_4 \;=\; \beta k_2\ V_A/V_F . \qquad (6.11)$$

Equation (6.11) allows equation (6.8) to be reduced to

$$M\ (t) \;=\; \frac{J}{V_A}\left(\frac{k_3 + k_4}{k_{12}\ k_3} - \frac{k_{12} - k_3 - k_4}{k_{12}(k_{12} - k_3)} \cdot e^{-k_{12} t} - \right.$$

$$\left. \frac{k_4}{k_3(k_{12} - k_3)}\ e^{-k_3 t} \right) . \qquad (6.12)$$

The four model parameters k_{12}, k_3, k_4 and V_A in equation (6.12), are related to the quantities in equation (6.9) by

$$k_3 = b_2 \ ,$$

$$k_{12} = b_1 \ ,$$

$$V_A = \frac{1}{B_2 b_2 + B_1 b_1} \ ,$$

$$k_4 = \frac{B_2 b_2 \ (b_1 - b_2)}{B_1 b_1 + B_2 b_2} \ , \qquad (6.13)$$

and can therefore be determined uniquely.

6.4 Estimation of Model Parameters

A set of model parameters is required, which provides the "best" fit of equation (6.7) to the experimental data. As gauges of the goodness of this fit, a number of criteria may be used. Of these criteria, the mean squared absolute error is used. By using this criterion it is assumed that the measurement errors are independent of the magnitude of the measurement. Although this assumption is not generally true for immunoassay, mean squared absolute error is shown later (section 6.7) to be an appropriate fitting criterion.

The mean squared absolute error E of the fit of the model to the experimental data values is defined as

$$E(k_{12}, k_3, k_4, V_A) = \frac{1}{N} \sum_{i=1}^{N} ([M](t = t_i) - [M_i^*])^2 \ , \qquad (6.14)$$

where $[M_i^*]$ is the measured value of ACTH concentration at time $t = t_i$, and N is the number of measured values.

The position in parameter space, defined by respective values of k_{12}, k_3, k_4 and V_A, of the minimum of E is determined using the "pattern search" algorithm, which is outlined in appendix 1.

The data from five experiments, performed on different sheep, are analysed to determine the values of the four model parameters. Results from this analysis are tabulated in table 6.4, together with the mean and standard error of the mean, for each of the four parameters.

Table 6.4 Model parameters obtained by fitting model to data.

	k_{12}	k_3	k_4	V_A	Mean Squared Error	Scaled r.m.s. Error %
Daphnis	0.069	0.0049	0.0084	3610	240	5.0
Cleo	0.060	0.0074	0.0113	3110	121	2.2
Samantha	0.160	0.0220	0.0240	1520	1084	7.8
Circe	0.056	0.0023	0.0086	4080	278	4.5
Carolyn	0.108	0.0079	0.0180	3730	107	3.5
SUMMARY						
Mean	0.091	0.0089	0.0141	3210		
Standard Error of the Mean	0.019	0.0034	0.0030	450		

6.5 Estimation of Biologically Active ACTH Concentration

Given the values of the two model parameters k_{12} and V_A, the concentration of biologically active ACTH [A] may be found as a function of time by solving equation (6.1). The general purpose simulation program SIMUL8 (described in chapter 7), is used to solve numerically equations (6.1), (6.2), (6.3) and (6.4), using the parameter values shown in table 6.4. These solutions, and the data from which the parameters are estimated, are shown in figures 6.5, 6.6, 6.7, 6.8 and 6.9.

It is evident in figures 6.5 to 6.9 that the secretion of cortisol more closely follows the concentration of biologically active ACTH predicted by the model, than the concentration of ACTH measured by immunoassay. The model was designed with this intention.

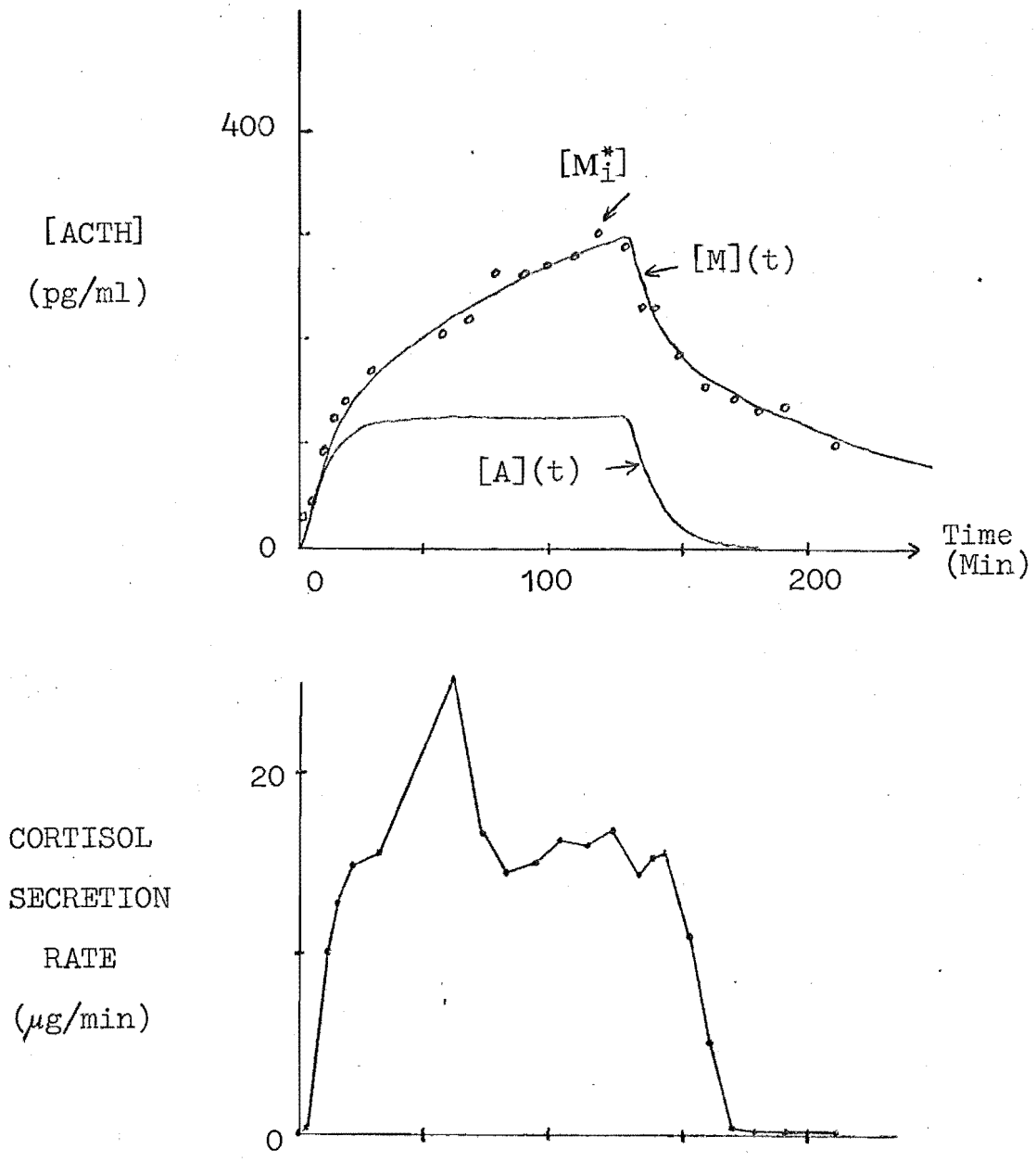


Figure 6.5 Carolyn sheep. Figure shows the immunoassay ACTH concentrations[M*], the predicted concentrations of biologically active ACTH [A](t), and the cortisol secretion rate.

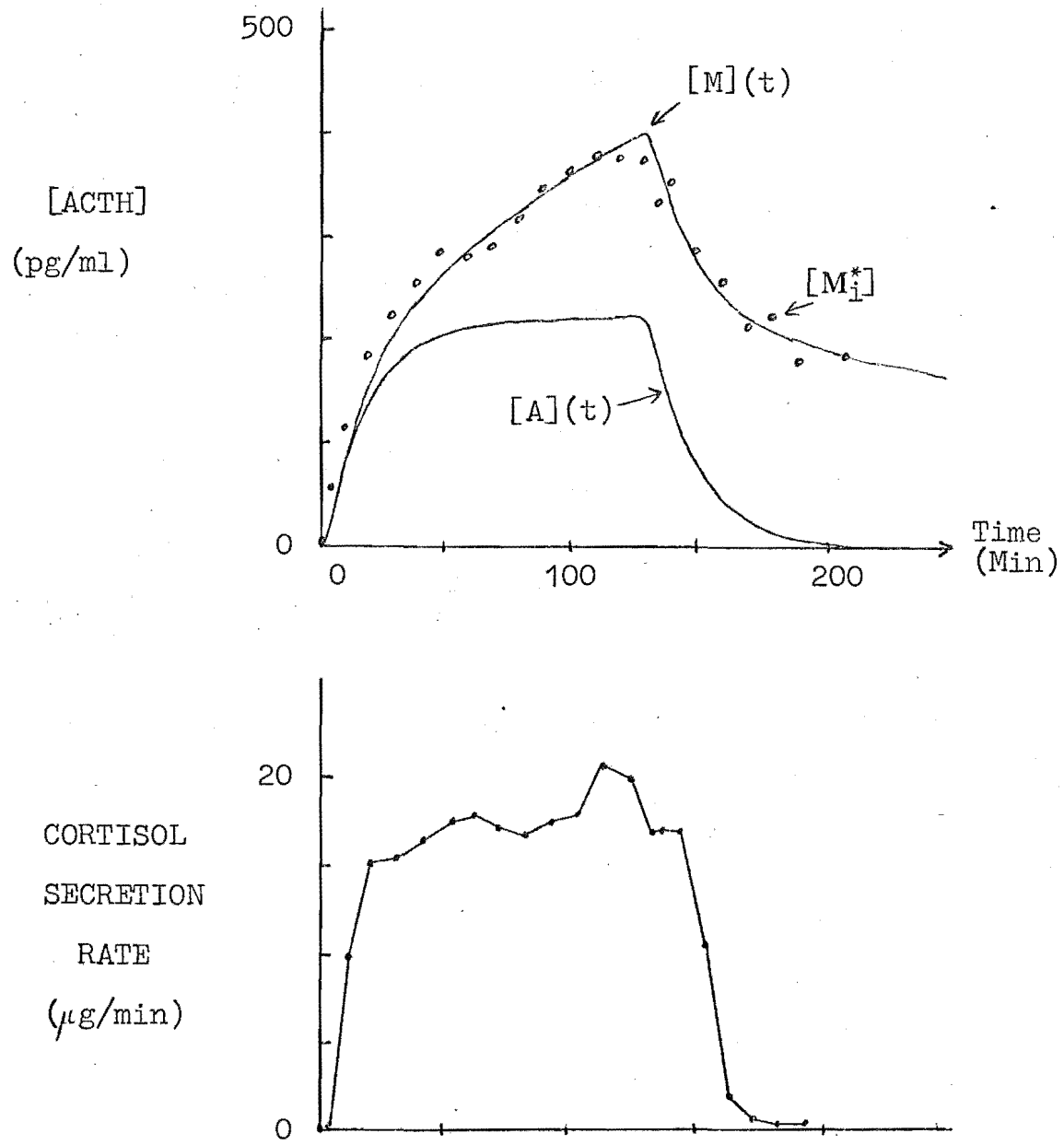


Figure 6.6 Circe sheep. For description of figure refer to figure 6.5.

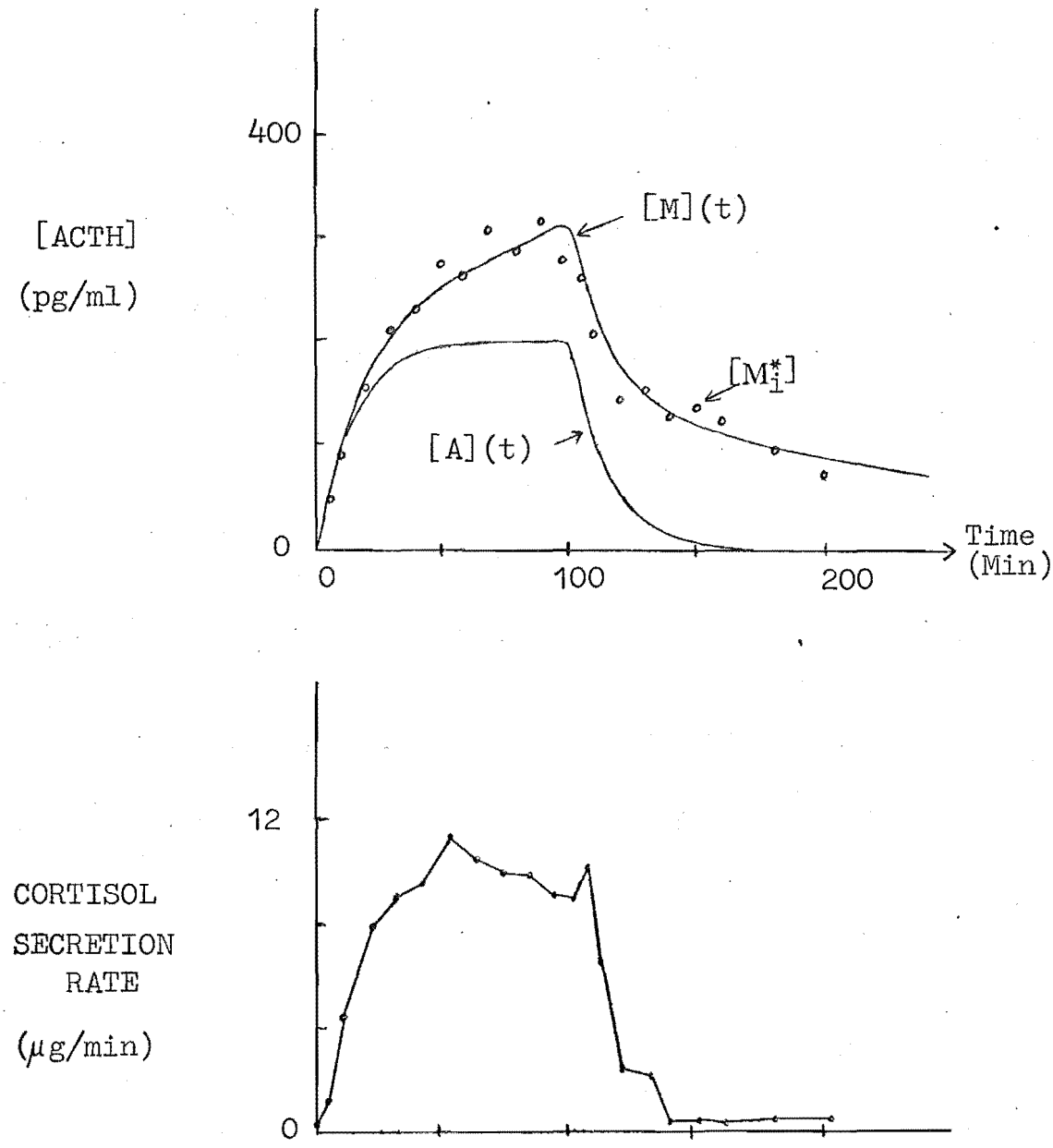


Figure 6.7 Daphnis sheep. For description of figure refer to figure 6.5.

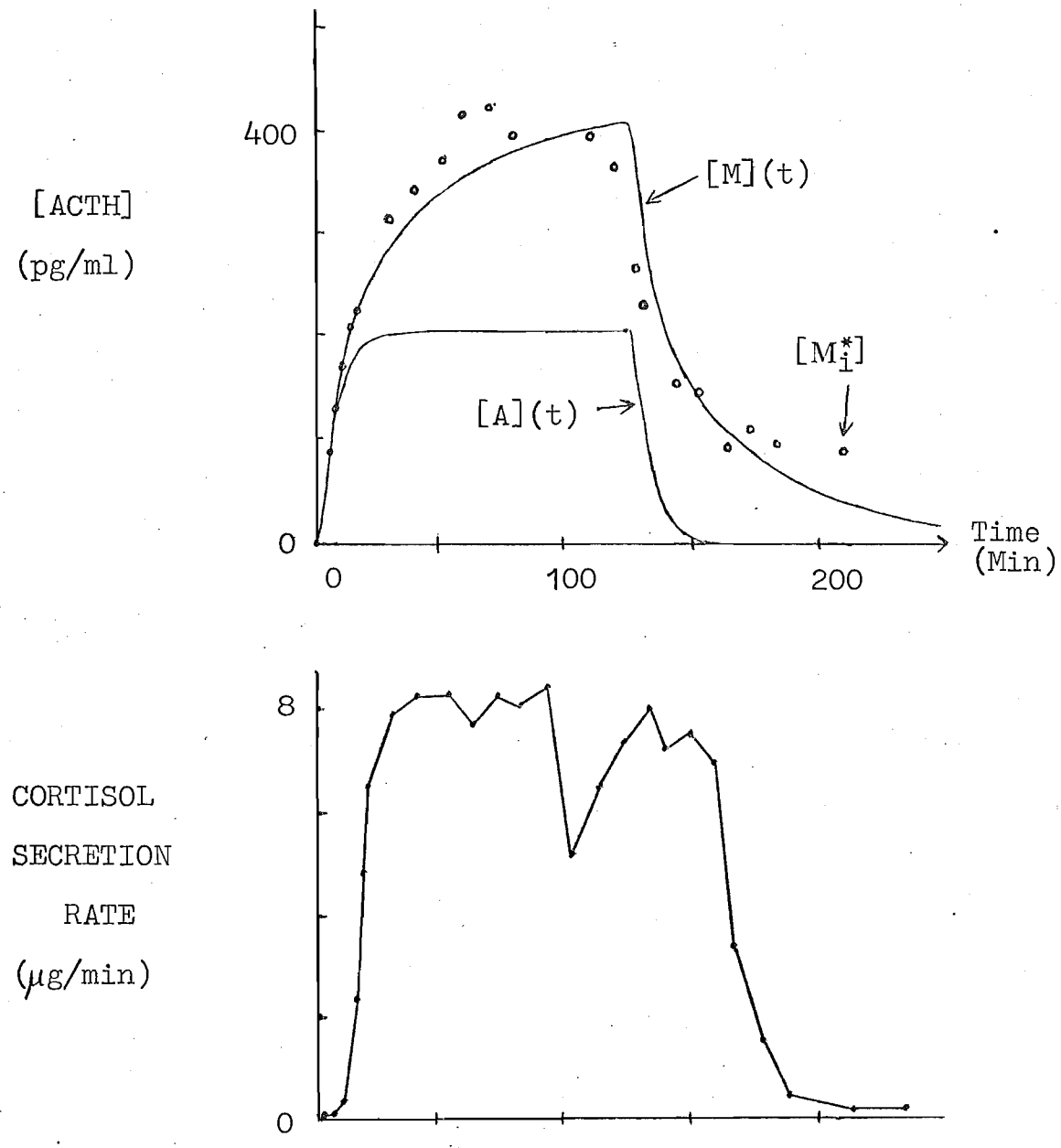


Figure 6.8 Samantha sheep. For description of figure refer to figure 6.5.

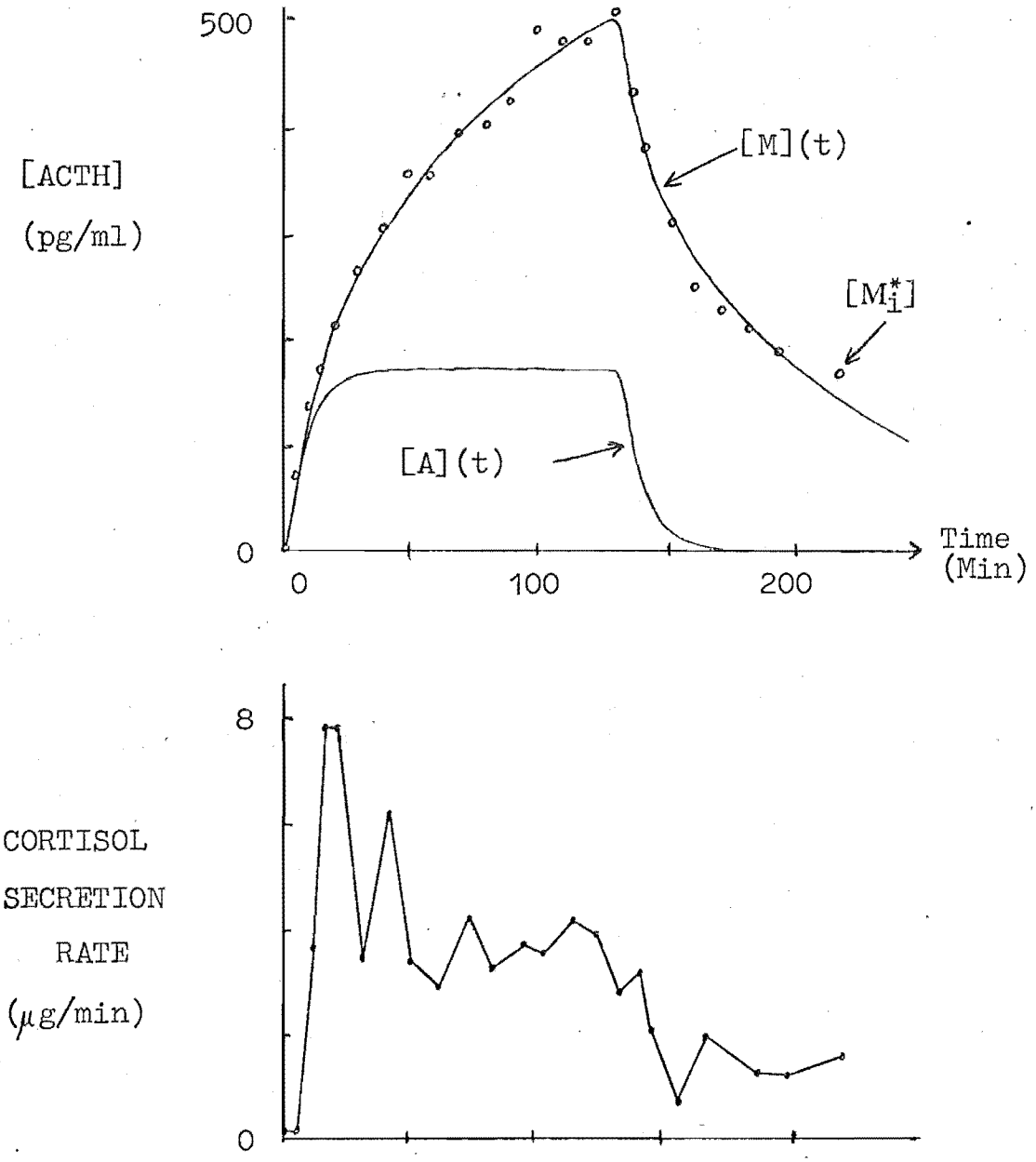


Figure 6.9 Cleo sheep. For description of figure refer to figure 6.5.

6.6 ACTH Distribution Volume

The volume V_A, over which the hormone ACTH is distributed, is the only one of the model parameters which can be related easily to a directly measurable part or process of the physical system. Because ACTH is probably small enough to escape through the walls of blood vessels, the volume of distribution of ACTH may be larger than the vascular volume of the animal.

As a check on the values for V_A obtained in section 5.4, the vascular volume of each animal was found by infusing RISA into the blood for a fixed period of time, and measuring the final concentration after the infusion was stopped (cf. section 5.4). The vascular volume found for each animal is tabulated in table 6.10. Also tabulated, are the ACTH distribution volume and the body weight for each animal and both RISA and ACTH volumes expressed in units of ml/kg body weight.

The figures in table 6.10 show that the distribution volume of ACTH is 60% greater on the average than the vascular volume measured by infusing RISA, and that there is a high degree of correlation between the ACTH and the RISA volumes.

Table 6.10 Results from RISA and ACTH distribution volume measurements.

ANIMAL	Body Weight (kg)	RISA Volume (ml)	RISA Volume ml/kg Body wt	ACTH Volume (ml)	ACTH Volume ml/kg Body wt	ACTH Vol/ RISA Vol
Daphnis	50	2200	44	3610	82	1.64
Cleo	44	1680	38	3110	71	1.85
Samantha	35	1510	43	1520	43	1.01
Circe	54	2700	50	4080	76	1.51
Carolyn	44	1830	42	3730	85	2.04
Mean	45.4	1980	43.4	3210	71.4	1.61
S.E. of the Mean	3.2	210	1.9	450	7.5	0.17

6.7 Discussion of Results

The concentrations of bioactive ACTH predicted by the model are in good agreement with the measured concentrations of cortisol secreted by the gland. This result makes the model, and the theory upon which it was based, a plausible explanation of the mechanism by which ACTH measurements are contaminated.

The high degree of correlation between the model parameters found for each sheep, and for V_A, the good correlation with volume measurements using RISA, are further indications of the plausibility of the model, although these results give no indication of the uniqueness of the model. This uniqueness, can in fact only be shown by eliminating all other possible models by experimental or other evidence.

The structure of the fragmentation model is the only two compartment linear structure capable of explaining the results. Thus, other model structures which could explain the results, must contain either nonlinearities, or more than two compartments. However, a number of theories of the reasons for the anomalies in ACTH measurement by immunoassay, reduce to the same two compartment linear structure of the fragmentation model. For example, it is unnecessary that the ACTH molecule is actually fragmented in the inactivation process; it need only be inactivated by a change of a chemical or physical nature, which renders it inactive to the steroid producing cells, without changing its immunological activity. The compartment F in figure 6.3, in this case describes the concentration of ACTH which is inactive with respect to steroid synthesis, while being

active immunologically. All of the analysis procedure described in the previous sections is unaffected by this change.

In section 6.4, the errors in the immunoassay measurement were assumed to be independent of the magnitude of the result. The validity of this assumption is tested by plotting the numerical difference between the data values and model solution, against the value of the model solution. If the errors are indeed functions of the magnitude of the result, this will be evident in the resulting "scatter diagram". In figure 6.11, the error versus amplitude of the result, for the data of Carolyn (cf. figure 6.5) is plotted.

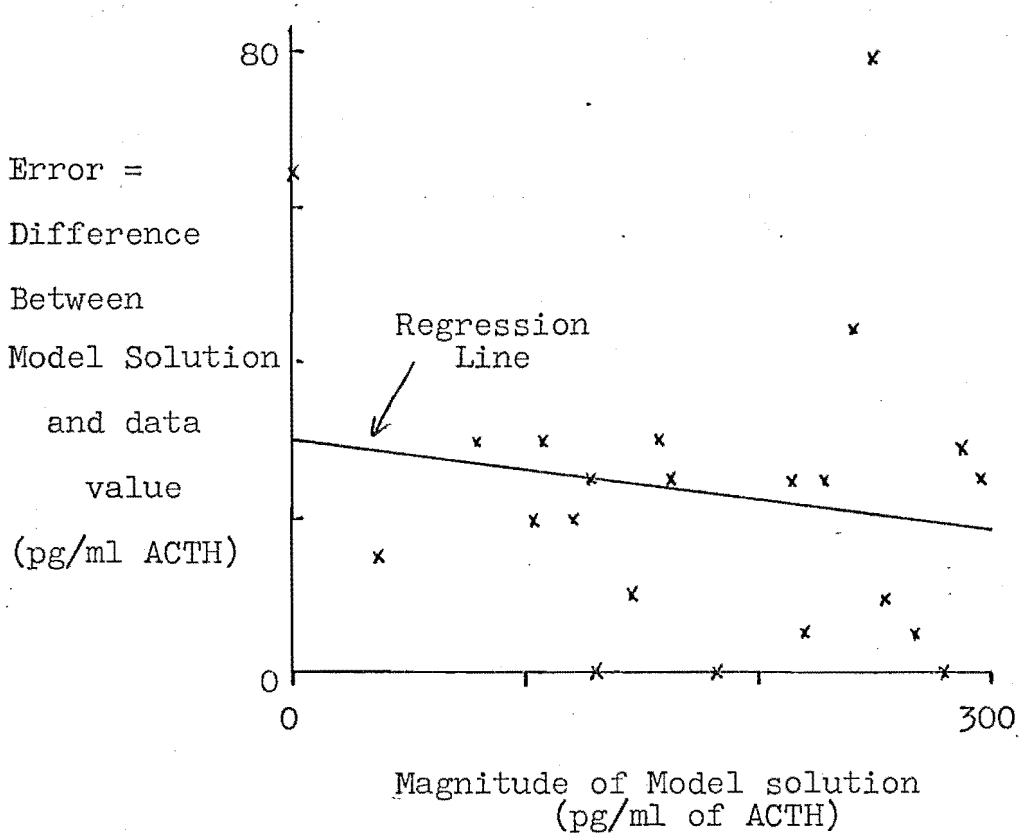


Figure 6.11 Curve fitting error as a function of data magnitude.

The linear regression line (cf. figure 6.11) shows that the measurement errors tend to reduce as the data magnitude increases, although the scatter of the data makes the significance of this result low. Similar trends are found in all five animals.

The method by which the parameters are estimated has a marked bearing on the results. This minimization technique, and its effects on the results are discussed in the next section.

6.8 Discussion of the Minimization Technique

The minimum of a function of a number of variables is described as the lowest value that function may attain when all possible combinations of the values of the variables are tested. In practice the time required to test all combinations of the variables precludes this approach. Computation time may be reduced by adopting a "search" procedure, in which regions of the variable space which reduce the function are explored. However, search procedures, which do not explore the entire variable space, are not guaranteed to converge on the global minimum (the true or absolute minimum) of the function and often converge on local minima (the minimum values in a region). There is no rigorous method of determining whether a local minimum is in fact the global minimum. By starting the minimization procedure from a number of different points in the variable space, different regions of that space may be explored, thus reducing the chance of mistaking a local minimum for the global minimum.

Two local minima were encountered when the data from the experiment on Cleo was used in the minimization procedure. The parameter and mean squared error values of the two local minima are shown in table 6.12.

Table 6.12 Comparison of model parameters at two local minima.

	1st Minimum	2nd Minimum
Parameter k_{12}	0.120	0.060
Parameter k_3	0.012	0.007
Parameter k_4	0.030	0.011
Parameter V_A	2500	3100
Mean Squared Error	165	121
Scaled Root Mean Square Error	2.5	2.2

While the root mean squared errors for the two local minima (cf. table 6.12) differ by only a small amount, the values of the four parameters are considerably different.
Figure 6.13 shows the data from the experiment on Cleo, and the model solution M(t) for the two minima in table 6.12. Whereas the fit of M(t) for the two minima are scarcely different, the model predictions for active ACTH concentration [A](t) show considerable variations between the two minima. Furthermore, the fact that two minima were found

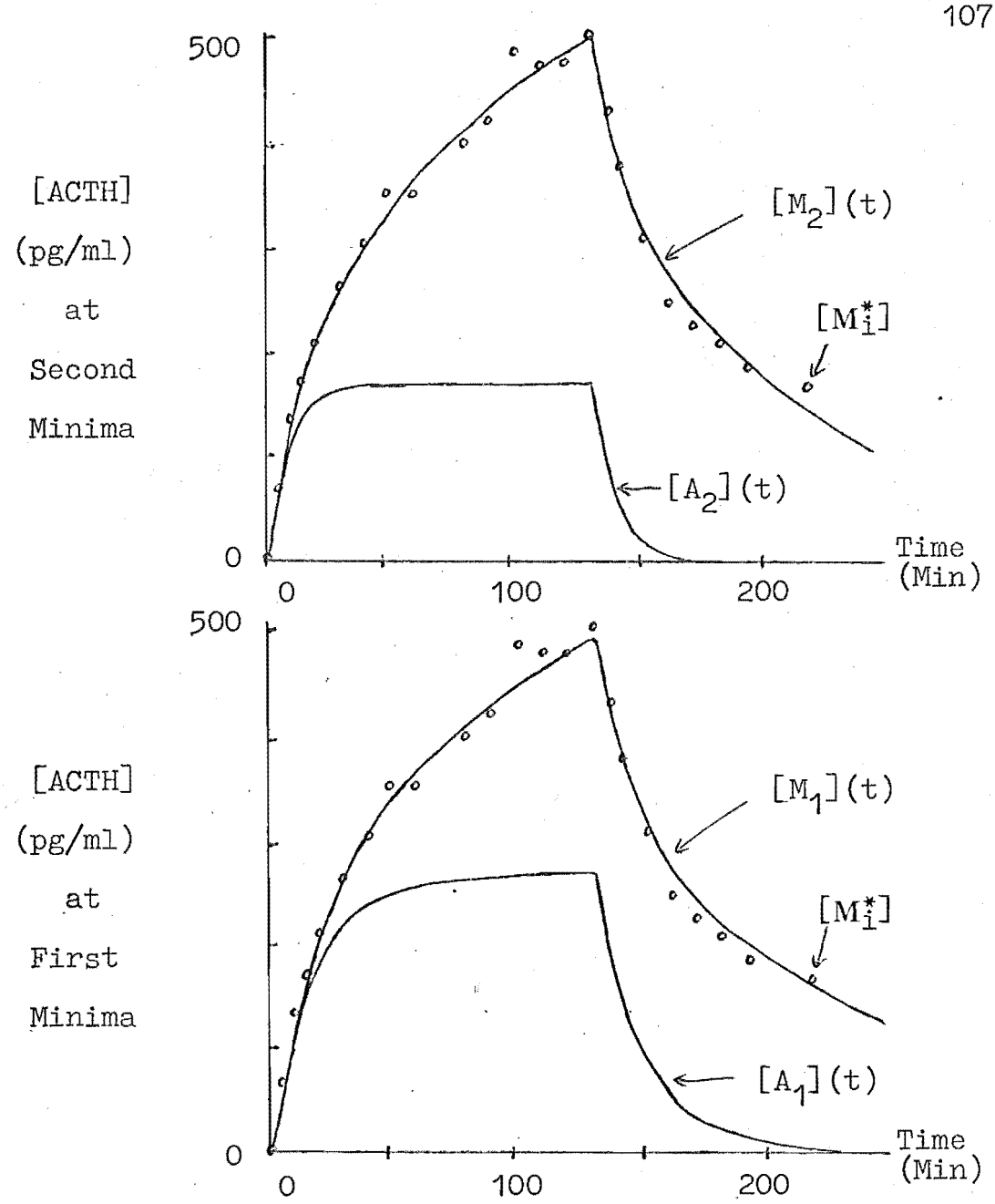


Figure 6.13 Model predictions of active ACTH concentration [A](t) at two local minima.

suggests that others with lower mean squared error values, may exist. This possibility may also be true for the analyses on the other four animals. For example, the mean squared error from the analysis of the data from Samantha (cf. table 6.4 and figure 6.8) is large, and the

corresponding parameter values are significantly different from those obtained in the other analyses. All attempts to locate another minimum with this data failed.

6.9 The Fragmentation Model Applied to Published Data

The validity of the fragmentation model may be tested by applying it to data in which both immunoassay and bioassay techniques have been used to measure plasma ACTH concentrations. Two such studies appear in the literature (Matsuyama et al., 1972; Bessar et al., 1971), the latter being very similar in experimental design to the experiments discussed in section 6.1

Using the immunoassay data of Bessar et al. (1971), which is shown in figure 6.14, the model parameters k_{12}, k_3, k_4 and V_A are determined by the procedure described in section 6.4. The parameter values obtained are listed in table 6.15.

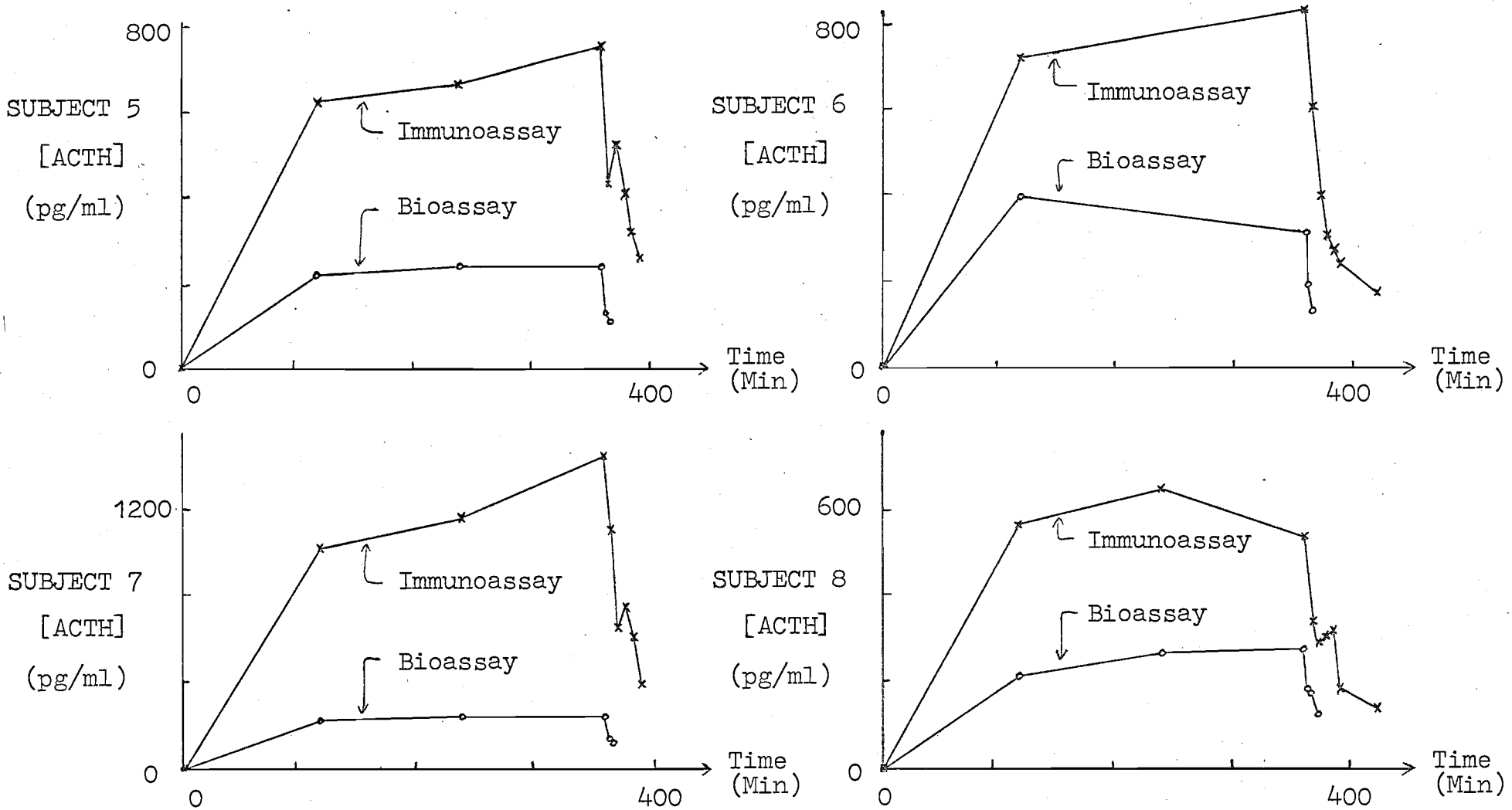


Figure 6.14 Immunoassay and bioassay ACTH measurements from Humans infused with ACTH. From Bessar et al. (1971).

Table 6.15 Model parameters obtained by fitting model to the data of Bessar et al. (1971).

	k_{12}	k_3	k_4	V_A	Mean Squared Error	Scaled rms Error %
Subject 5	0.454	0.019	0.054	2050	2156	6.2
Subject 6	0.084	0.0005	0.0008	3000	203	1.7
Subject 7	0.090	0.9×10^{-5}	0.0014	2020	7839	6.1
Subject 8	0.363	0.018	0.028	2000	1530	6.0
Summary of Bessar's parameters						
Mean	0.248	0.0094	0.021	2270		
Standard Error of the Mean	0.095	0.0053	0.013	244		
Summary of sheep parameters (from Table 6.4)						
Mean	0.091	0.0089	0.0141	3210		
Standard Error of the Mean	0.019	0.0034	0.0030	450		

Figure 6.16 shows the model solutions M(t) and A(t), which are found by the procedure described in section 6.5, compared with the data from four of the experiments from Bessar et al. (1971). The model predictions for biologically active ACTH compare well with the measured concentrations of this moiety for subjects 5 and 8, however, the concentrations for subjects 6 and 7 do not compare well.

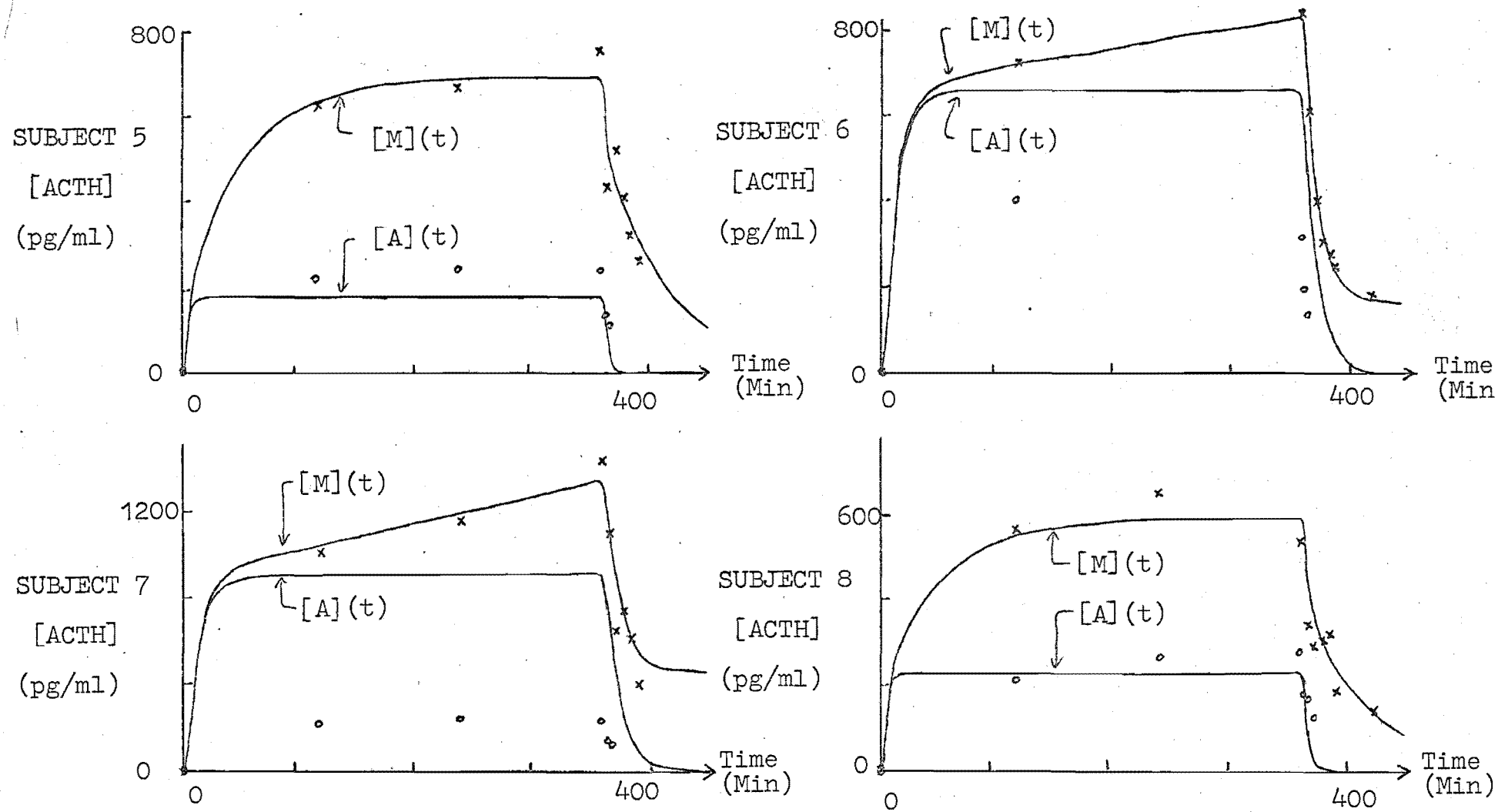


Figure 6.16 Data values and model predictions where both bioassay and immunoassay ACTH measurements are available (cf. Bessar et al., 1971).

Inspection of table 6.15, shows that three of the parameters (k_3, V_A and k_4) are similar in magnitude to those found for the sheep, while the parameter k_{12} is considerably different for the two species studied.

The parameter values obtained from the data of Bessar et al. (1971) show considerably more "within parameter" variation than the corresponding parameters for the sheep. This variation is possibly caused by the small number of data points used in this analysis, i.e. for Bessar's data only 9 or 10 data points are available, whereas in the sheep studies there are 21 to 24 points.

6.10 Measurement of Endogenous ACTH Concentrations

The technique described in sections 6.4 and 6.5 for estimating the concentration of biologically active ACTH, requires knowledge of the ACTH infusion I(t) (cf. equation 6.1). In many experimental situations the source of ACTH is the pituitary gland and the time course of this secretion is unknown. ACTH concentration, measured by immunoassay, is still subjected to contamination by biologically inactive ACTH, when this hormone reaches the blood from the pituitary.

By using the fragmentation model (section 6.2) it is possible to estimate the concentration of biologically active ACTH from the immunoassay data. Equations (6.2) and (6.3) describe the relationships between the three ACTH moieties (A, F and M) shown in figure 6.2.

Eliminating [F] from equations (6.2) and (6.3) gives

$$[\dot{A}] = -(k_3 + k_4).[A] + [\dot{M}] + k_3.[M] \qquad (6.15)$$

To solve equation (6.15), the derivative of the data, i.e. $[\dot{M}]$, must be determined. The ACTH data contains noise in the form of measurement and sampling errors. Because the process of differentiating data amplifies any noise which is present in that data, it is desirable that the term $[\dot{M}]$ in equation (6.15) is eliminated. From equation (6.3)

$$\beta[F] = [M] - [A] , \qquad (6.16)$$

and hence

$$\beta[\dot{F}] = [\dot{M}] - [\dot{A}] , \qquad (6.17)$$

so that equation (6.15) becomes

$$\beta[\dot{F}] = -(k_3 + k_4).\beta[F] + k_4[M] . \qquad (6.18)$$

Thus, providing the values of k_3 and k_4, and a functional form for [M] may be found, equation (6.18) may be solved by numerical integration to yield $\beta[F]$, from which [A] may be determined using

$$[A] = [M] - \beta[F] . \qquad (6.19)$$

The two parameters k_3 and k_4 must be determined from independent experiments on the animal, as they are not evaluated in this procedure. The blood samples, from which ACTH concentration is measured by immunoassay, are taken at intervals of about ten minutes. Estimates of the ACTH concentration between adjacent samples, are found by

interpolating the data with a smooth curve. The interpolation method of Akima (1970), which is described in Appendix 2 is used to interpolate the ACTH data, thus providing a functional form for [M] which may be used in the solution of equation (6.18).

6.11 Application of the Model to the Measurement of Endogenous ACTH

The experimental data, resulting from measurements made by the Lincoln group, of both ACTH concentration and cortisol secretion rate in sheep, were analysed by the technique described in the previous section. The cortisol secretion rates (shown in figure 6.17) do not correlate with the immunoassay ACTH concentrations (i.e. when cortisol is secreted at low rates the ACTH concentration remains high). The poor correlation is probably due to biologically inactive ACTH interfering with the radio-immunoassay measurements.

The model parameters (k_3 and k_4) of equation (6.18) used to determine biologically active ACTH concentrations are the mean values of those parameters from table 6.5. Figure 6.16 shows the biologically active ACTH concentration, estimated from the immunoassay ACTH data using equations (6.18) and (6.19). The estimation is performed using SIMUL8 (cf. chapter 7). The predicted ACTH concentration shows a better correlation with the cortisol secretion rate curve than the original immunoassay ACTH results. When the cortisol secretion rate reaches low values the predicted

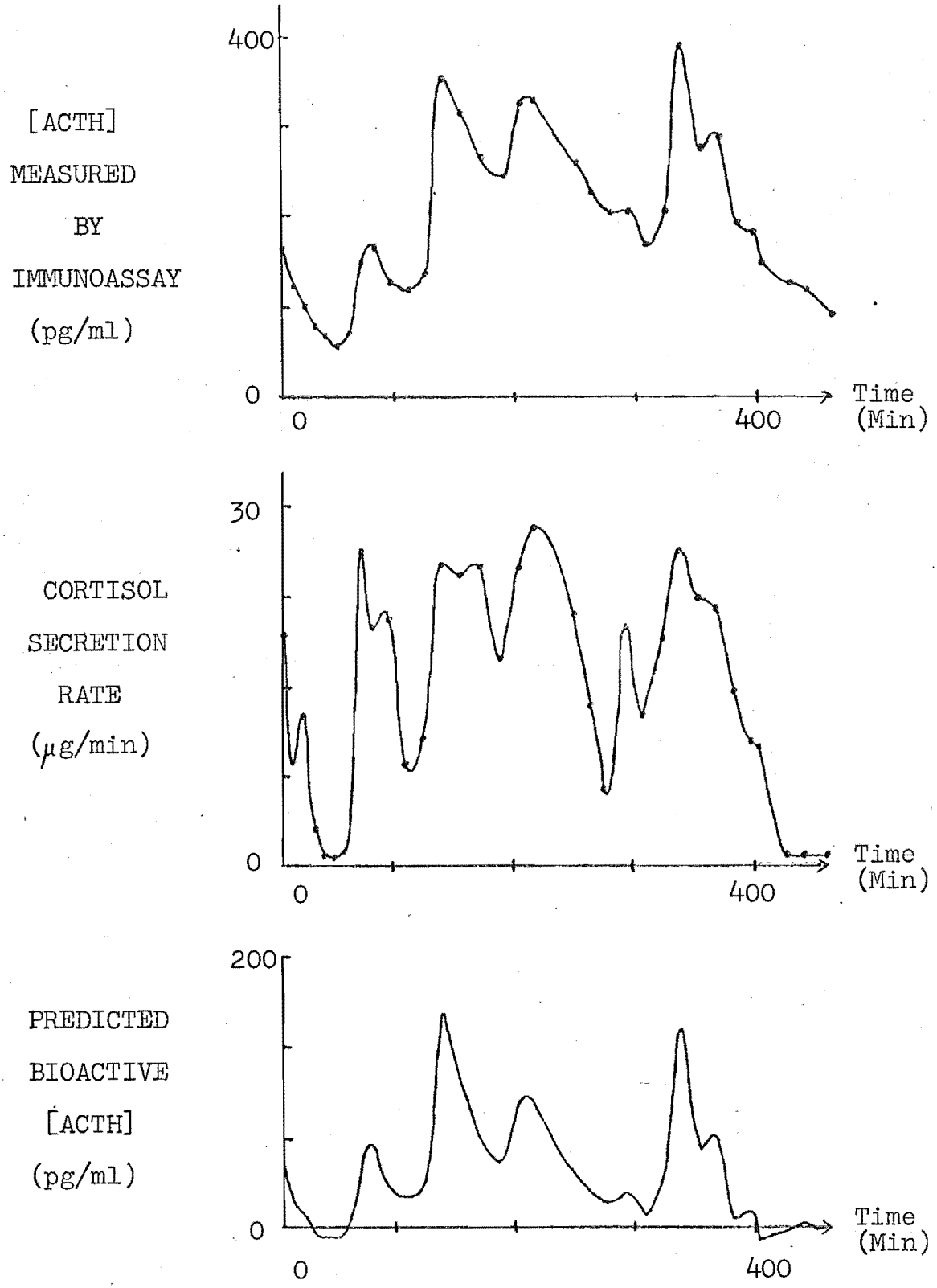


Figure 6.17 Bioactive ACTH concentrations predicted by the model from experiment where the source of ACTH is the pituitary.

ACTH concentration is also low. In fact, in two places the predicted ACTH concentration drops below zero, which is possibly due to small errors in the original data.

6.12 Summary

Measurements of ACTH using immunoassay show unexpectedly high ACTH concentrations when compared with bioassay measurements.

Bessar et al. (1971) explain the difference between the two techniques by postulating that the ACTH molecule is physically fragmented - the fragments retaining immunoreactivity but not bioactivity. A two compartment model of the process described by Bessar et al. is capable of mimicking the measured response of ACTH using bioassay and immunoassay, provided that the rate of breakdown of ACTH fragments is slower than that of the entire ACTH molecule.

Using the model developed in this section, it is possible to estimate the concentration of biologically active ACTH from the immunoassay data, where ACTH is infused into the animal, provided sufficient data are available. However, the estimation process is not simple. It involves least squares parameter estimation which does not always converge onto the true parameter values and can produce erroneous results.

Where ACTH is not infused into the animal, but rather is secreted ad lib. by the pituitary, it is possible to predict bioactive ACTH concentrations from immunoassay data provided suitable values for two of the model parameters can be obtained.

The model developed in this chapter does not in fact rely on the ACTH molecule being fragmented. Any process which inactivates the molecule in a biological sense, but does not affect its immunoreactivity, can be explained by the model in its present form.

6.13 Discussion of ACTH Immunoassay

The proposed model of ACTH inactivation (or fragmentation) provides a plausible explanation, both analytically and physiologically, of the anomalies which occur when ACTH is measured by radioimmunoassay. While the analysis techniques presented in this chapter allow estimation of true ACTH concentrations from immunoassay data, they indicate that use of ACTH immunoassay in normal clinical practice, where few samples are analysed, is dangerous. Two to five times the true ACTH concentration can be indicated by ACTH immunoassay (cf. figure 6.16). Even on a comparative basis ACTH immunoassay results can show large variations. Note that subjects 7 and 8 in figure 6.16, who have comparable immunoassay ACTH concentrations, show bioassay ACTH concentrations of 120 and 270 pg/ml respectively.

Modifications of the immunoassay technique may provide a better method of determining bioactive ACTH concentrations. An antibody, which binds more specifically to the bioactive portion of the molecule, would reduce the contamination of the immunoassay results by biologically inactive ACTH. Alternatively, the "sandwich technique", developed by Hales (1971) for the immunoassay of insulin, may provide

a more specific assay for ACTH. This technique uses two antibodies which bind to different parts of the molecule; both must be present before the molecule is detected. The sandwich technique would only be useful if the inactivation process involves physical fragmentation of the ACTH molecule.

CHAPTER 7

SIMUL8 - A SIMULATION PROGRAM

Simulation is the process of solving the algebraic and differential equations which describe a model. A continuous simulation technique (cf. section 2.2) was required for this research, and for the research of McKinnon (1973). Because the level of interaction provided by the simulation systems available to us was insufficient, the development of a flexible simulation system was undertaken as a joint project. This chapter describes the main features of the resulting program, (named SIMUL8) and the philosophy upon which it is based. A detailed description of the operation of SIMUL8 is given in appendix 4, which is in the form of an operator's manual (this was written jointly by the author and Dr A.E. McKinnon).

7.1 Background to SIMUL8

When the need to solve the equations of a model was first met, a number of simulation systems were investigated, to determine which would be the most convenient to use. The analogue computer, while capable of solving the equations rapidly, was found to be slow to set up. Moreover, changes made to the model structure on the analogue computer took considerable time to implement. The labelling and documentation of simulation results must be added by hand to graphical output from the analogue computer, a process which is prone to error. CSMP (Brennanand Silberberg, 1968), a digital

program which simulates the analogue computer, provides better documentation of the results, and may be set up faster than the analogue computer. DSL (Syn and Linebarger, 1966), a simulation program in which the model is described as differential equations, entered to the computer as FORTRAN language statements, provides the same advantages as CSMP. In the CSMP program the set up procedure occurs in two phases, translation of the model into algebraic and differential equations, and translating these equations into the basic analogue computer elements. In DSL this latter phase is eliminated.

Both DSL and CSMP are operated in a batch mode, each simulation being submitted to the computer as a job, which is collected a number of hours later. Program errors may then be corrected, or modifications to the model structure or the model parameters made, and the job resubmitted. This process can be both time consuming and frustrating, particularly when a new model is being "debugged". Where access to a dedicated computer or a terminal of a time shared computer is available the procedure of model investigation can be dramatically improved. Errors can be quickly found and corrected, different parts of a model quickly explored, and graphical and numerical output from the computer minimized (i.e. it is not necessary to plot or list the values of every variable in case they are needed, as it is quicker to rerun the simulation looking at other variables if they are found to be of interest).

The EAI 640 computer in the Electrical Engineering Department offered us an ideal environment in which to develop

an interactive simulation language, because it is not operated in a batch mode, and because it incorporates a visual display unit suitable for graphical and numerical output of information at high speed. The advantages offered by interactive modelling, which were described above, were thought to justify the time required to develop a new simulation language.

7.2 SIMUL8 Philosophy

Of the two commonly used methods of describing a model on the computer (viz. equations and function blocks), the equation form used in DSL was chosen as the most convenient. The reasons for this choice are

. Standard programs exist in the computer library to convert FORTRAN language statements into machine code.

. Manipulation of equations can be performed quickly and conveniently using a "text editing" program available in the computer library.

. The equations of the model may be translated easily into a form suitable for the simulation program (cf. section 7.3).

. The FORTRAN language, in which the model equations are written, is well documented and in general most users are familiar with it. In contrast, it is necessary to completely define a new language when a block oriented structure is used.

. The FORTRAN language provides considerable flexibility, e.g. function subprograms may be easily incorporated to provide the user with access to such functions as SIN, SQRT etc. (cf. section 7.5).

The equation oriented languages suffer from one disadvantage. It is generally necessary to pass the language statements through a precompiling phase in which the order of the equations is manipulated until all variables are assigned values before they are used in subsequent calculations. These "sort" algorithms (cf. Stein and Rose, 1960) are difficult to implement if the complete flexibility of the FORTRAN language is to be retained. An alternative to precompilation was found, in which the equation order is checked at "run time", any errors found being flagged and left for the user to correct with the aid of the text editing program.

Each of the basic operations of the simulation (e.g. plotting, changing parameters and solving the equations) is assigned a two character mnemonic. By entering these mnemonics into SIMUL8 via the teleprinter keyboard, the appropriate operations are started. On completion of the operation a new command may be entered using its mnemonic. In this way the user is given complete control over the progress of his model investigation.

In our opinion, a simulation program based on the ideas described in this section provides the modeller with all of the advantages offered by DSL. Further, considerably more flexibility in the model investigation is provided by using the advantages of an interactive computing environment.

7.3 Software Organization

The simulation program (SIMUL8) is implemented on the EAI 640 computer. The program is coded using both FORTRAN and ASSEMBLY languages and for this reason it is machine particular in design, but not in concept.

The mainline program of SIMUL8 accepts commands from the teletypewriter keyboard, in the form of a two character mnemonic describing the operation to be executed. Each command mnemonic has associated with it a utility subroutine which performs the operation defined by the mnemonic. The SIMUL8 mainline thus decodes the mnemonic and executes the appropriate utility subroutine.

The most important of the SIMUL8 utility subroutines solves a set of algebraic and differential equations provided by the user. These equations are supplied to SIMUL8 in the form of a FORTRAN subroutine, called DIFEQN, which describes the first derivatives of the model variables as functions of the model variables and parameters. DIFEQN may be constructed using most of the standard FORTRAN conventions, including calls to the standard functions for trigonometrical and algebraic relations, such as SIN and SQRT, and to special functions written for SIMUL8, for example, step and ramp functions, negative and positive clippers, relays and hysteresis modules.

Time is incremented by SIMUL8, and at each time step the subroutine DIFEQN is executed and the values of the model's dependent variables are updated by an Adams-Bashforth (Bashforth and Adams, 1883) predictor formula. A second order formula proposed by Henrici (1962) from the original general order algorithm is used. The updating procedure is

$$y_{n+1} = y_n + \tfrac{1}{2}.H.(3.\dot{y}_n - \dot{y}_{n-1}) , \qquad (7.1)$$

where y_n is the solution at time

$$T = (n - 1)H , \qquad (7.2)$$

H is the discrete time step, and

$$\dot{y}_n = \frac{d}{dt} y_n . \qquad (7.3)$$

Because equation (7.1) requires the derivative of y at the previous step (viz. $\dot{y}_{n-1}$), the initial step must be made using a different method. The Euler integration formula (cf. Noble, 1964)

$$y_1 = y_0 + H.\dot{y}_0 \qquad (7.4)$$

is used for the first step.

The model variables, their derivatives, and the model parameters are transferred between SIMUL8 and DIFEQN via FORTRAN common blocks named EQTNS, DERIV and MODPAR respectively. An example of the DIFEQN subroutine used to simulate the ACTH fragmentation model (cf. section 6.3) is listed below.

```
      SUBROUTINE DIFEQN                                    001
      COMMON/EQTNS/A,F,M                                   002
      COMMON/DERIV/DA,DF                                   003
      COMMON/MODPAR/VA,VF,K1,K2,K3,BETA                    004
      REAL M,K1,K2,K3,I,A,F,M,DA,DF,VA,VF,BETA,T           005
      T = TIME (NOW)                                       006
      DA = I(T)/VA - (K1 + K2)*A                           007
      DF = K2*VA*A/VF - K3*F                               008
      M = A + BETA*F                                       009
      RETURN                                               010
      END                                                  011
```

(7.5)

The variables A, F and M declared in line 2 are the three model variables, DA and DF are the time derivatives of A and F respectively, and line 4 lists the six model parameters. Line 5 declares all of the variables and parameters of the subroutine to be floating point numbers (all operations within SIMUL8 are executed in floating point mode). Lines 7, 8 and 9 describe the model equations. These equations should be compared with equations (6.1), (6.2) and (6.3). (Note that DA and DF represent $\frac{d}{dt}A$ and $\frac{d}{dt}F$ respectively.) The one to one correspondence between the model equations and DIFEQN suggests the ease with which this subroutine is coded.

During the course of a simulation the model variables are sampled automatically by SIMUL8 and the resulting solution values are saved for subsequent inspection. Up to four model variables may be saved in this way. When a simulation is completed, the variables which have been

sampled, may be plotted against time on the display screen, using one of the four plotting commands available (see table 7.1).

The values of the model parameters, and the initial values of the differential equation solutions, may be interrogated by the user at any time, and new values entered through the teletype.

The three operations described (viz. simulation, plotting and changing model parameters) are the most important utilities contained in SIMUL8. They enable most of the processes involved in investigating models to be undertaken interactively. A number of other utilities which are provided allow the user to manipulate the parameters of the integration algorithm, transfer information between the computer and paper tape, and perform various manipulations on the model solutions. Table 7.1 shows all of the utilities available in SIMUL8, and the two character mnemonics by which they are accessed.

Table 7.1 List of SIMUL8 commands.

Utility Mnemonic			Utility Function
SS			Start simulating the model.
MP	A:	≠	Interrogate model parameters and enter new values.
YØ	A:	≠	Interrogate initial conditions and enter new values.
DA	B,C,D	≠	Erase screen and plot variables against time with automatic scaling.

Utility Mnemonic			Utility Function
DF	B,C,D	≠	Erase screen and plot variables against time using the same scale factor for all variables.
DO	B,C,D	≠	Plot variables on the current graph using scale factor of the preceding curve drawn.
DN	B,C,D	≠	Plot variables on the current graph using automatic scaling.
DT			Define new step length for integration algorithm.
TF			Define new time to terminate simulation.
TR			Define time to halt simulation so that solutions may be investigated, i.e. TRACE solution.
RS			Restart simulation after a halt by TR command.
YY	A:	≠	Interrogate solution of model variables at current time, e.g. the time the simulation is stopped by TR command.
ED			Read data from papertape. This could be experimental data to be compared with model solution.
LD			Output the values of a sampled variable to the teletype, display screen or papertape punch.
RY			Read new initial conditions from papertape.

Utility Mnemonic	Utility Function
OY	Output initial conditions to paper-tape punch.
RP	Read model parameters from papertape.
OP	Output model parameters to papertape punch.
CT	Allow users comments or messages to to be typed on teletype.
CD	Allow users comments or messages to be typed on display screen.
ME	Calculate mean squared error between experimental data (ED) array and equation solution.
CM	Save the COMMON blocks on disc, execute the text editing program so that the user can change his model statements, compile the new DIFEQN subroutine, reload the program and the COMMON blocks, and continue.
MO	Terminate the use of SIMUL8.
NE	Define the number of differential equations.
NS	Define the maximum number of integration steps.
SA	Save the samples of an equation solution for later inspection. N.B. Normally these samples are lost when the model is next simulated.

Utility Mnemonic	Utility Function
PR	Define the model variables which are to be sampled for plotting during the course of a simulation.
EY	Exchange the values of the initial conditions and the equation solutions. This enables the user to run his model until a steady state is reached and to then set the initial conditions to the steady state values.
UU	Executes a user supplied FORTRAN utility program to perform auxiliary calculations.

Note ≠ User types a number (A) describing the position of the parameter in COMMON.

≠ User types the numbers (B,C,D) of the variables to be plotted.

7.4 Operation of SIMUL8

A number of system programs are used in conjunction with SIMUL8. These include the FORTRAN compiler, and a program which allows small changes to be made to source statements using the visual display screen (i.e. a text editing program). The execution of these programs and SIMUL8 is automatically controlled by special executive programs. The following sequence occurs when SIMUL8 is used to solve a set of model equations.

The user prepares his DIFEQN subroutine and punches it onto papertape (this is usually performed off-line). Once on the computer the SIMUL8 executive programs load into core memory, and execute the text editing program, thus allowing the user to make small corrections to his source statements. On the completion of the editing, the source statements are automatically compiled by the FORTRAN compiler and then loaded into core memory along with the main body of SIMUL8. Following the execution of SIMUL8 the user is asked to enter the values of the model parameters, initial values of the differential equation solutions, and various parameters used by the integration algorithm and plotting programs. When this initialization phase is complete, the user may employ the standard SIMUL8 utilities to solve model equations, graph the results and perform various other manipulations described in table 7.1.

If at any time the user wishes to change the structure of the model, a process requiring modification of the source statements of DIFEQN, the change model utility (CM) is executed. This causes the contents of all COMMON blocks to be saved on the disc and control to be automatically transferred to the text editing program. The statements within DIFEQN may then be changed to describe a new model. Following editing, the new DIFEQN subroutine is compiled and loaded into core memory with SIMUL8. The initialization procedure is now bypassed; instead the COMMON blocks are reloaded into core from the disc and the user is immediately given access to the SIMUL8 utilities. The "change model"

procedure may be repeated as often as desired. The user can therefore modify model equations as well as model parameters in the course of the model investigation.

7.5 Advanced Features

When a set of model equations is prepared for a simulation package such as SIMUL8, there is a danger that some of the equations are placed in an incorrect order. This results in a variable being used in a calculation before it has been assigned a value. As a result of this the accuracy of the solution can be degraded and the affected variable plotted incorrectly. Simulation languages such as DSL (Brennan and Silberberg, 1968), check the order of the equations in the source statements before these statements are compiled. If an error in the equation order is detected by DSL, the equation order is changed. To write the software for such a precompilation check was clearly outside the scope of the SIMUL8 programmers, within the time available. A scheme was devised which allowed the equation order to be checked at run time. Any changes required to the order of the equations are then performed by the user. This scheme relies on three features of the EAI 640 system programs. These are

(a) The FORTRAN compiler does not assign values to any of the variables in a FORTRAN program. When the program is loaded into core, all variables which are not predefined in DATA statements will take on values contained in the memory before the program was loaded.

(b) All arithmetic operations are performed by special subroutines which are loaded into core with the FORTRAN program.

(c) All floating point numbers are stored in core memory in a normalized form, i.e. the numbers are stored in two parts A and B where

$$\text{Number} = A * 2^B , \tag{7.6}$$

and

$$\tfrac{1}{2} \leq A < 1 . \tag{7.7}$$

Any number which violates equation (7.7) is said to be unnormalized.

The detection of equations that are out of order is implemented as follows. Prior to loading SIMUL8 and the DIFEQN subroutine into core memory, all available memory cells are assigned an unnormalized value. When SIMUL8 is loaded, all memory locations will therefore contain unnormalized numbers. The arithmetic routines are modified so that all numbers which are found to be unnormalized will have their storage addresses typed on the teletypewriter. This has the effect that all variables which are not defined before they are used, will be flagged as errors at run-time. This method of detecting equations which are out of order, is simple to implement and effective. An additional advantage is that other error conditions are also detected. These include equations missed, and in some cases mispelt variable names (e.g. the similarity between I and 1, and O and Ø often cause errors).

Auxiliary calculations which are to be executed during the course of a model investigation may be performed in a user supplied subroutine named UUPROG. This program is written in FORTRAN and is loaded into core memory with DEFEQN and SIMUL8. By typing the command UU the user may cause the subroutine UUPROG to be executed. UUPROG can be put to a variety of uses. Statistical studies on model solutions, and resetting the base number of a random number generator are examples. UUPROG, like DIFEQN, may use most of the language statements available in FORTRAN, including calls to standard function programs and user supplied subroutines.

7.6 Conclusions

The simulation program SIMUL8 provides the modeller with a convenient tool for investigating the operation of small models of a variety of systems. The interactive features of SIMUL8 enable the user to test his model swiftly (e.g. a simple model, involving say two differential equations, could be coded and prepared for the computer within an hour, and simulation results be available to the user after an hour on the computer). Furthermore, SIMUL8 provides good documentation of the modelling efforts. It was found that the interactive features of SIMUL8, allow it to be used more effectively than the simulation languages DSL and CSMP. The documentation of the results is significantly better than that provided by analogue computation and is at least as good as that provided by DSL and CSMP.

The translation between the equations which describe the model, and the FORTRAN subroutine DIFEQN, used by SIMUL8, is simple, due to a close correspondence between the two. This makes the process of setting up a model rapid.

SIMUL8 has to date been used to investigate a number of models developed by users with quite different interests. These include models of the water and energy balance of a leaf, a bridge rectifier with inductor-capacitor filter, a thyristor controlled AC motor, and the digestive system. All modellers have found SIMUL8 simple to learn and they have reached a stage of satisfactory competence within a few hours of using the language.

The simulation program was invaluable when the models discussed in chapters 4.0 and 5.0 were being developed and investigated. The program also enabled models developed by others, such as those discussed in section 3.6, to be tested and modifications to them tried. The effort involved in implementing the models of section 3.6 on the analogue computer, or using DSL, would not have been justified.

CHAPTER 8

MAGNETIC TAPE STORAGE SYSTEM

The bulk storage available on the Electrical Engineering Department's EAI 640 computer was insufficient to allow large programs such as SIMUL8 (see chapter 7.0), to be permanently resident without causing considerable inconvenience to other computer users. A magnetic tape storage system was designed and built to alleviate these storage limitations.

Historically, the magnetic tape project started in 1969 when I was completing my third professional year for a B.E. In collaboration with two other students (McKinnon and Lowinger q.v.) I undertook the design of a simple magnetic tape unit (Jordan, 1969). Because of the relatively short time available, this unit was never completed. Throughout the period that this initial unit was developed, the requirements of the magnetic tape underwent dramatic changes through variations in the patterns of use on the computer.

During the vacation, prior to starting the research described in this thesis, I spent four weeks laying down the basic design of the tape controller described in this chapter.

Later, when the hardware for the controller had arrived I realized the importance of the tape unit to the furthering of my research and devoted time to the completion of the design, overseeing the construction, testing, and development of suitable software for easy use of the magnetic tape.

The software and much of the testing was completed in conjunction with I.L. Roxbourgh.

In this chapter the hardware, the software, and the implementation of the magnetic tape storage unit are described.

8.1 Need for and Choice of a Bulk Storage System

The EAI 260 fixed head disc, which is implemented in the Electrical Engineering Department, has sixty four information tracks, each with a storage capacity of less than six thousand sixteen bit data words. System programs, for example the compilers and the utilities, occupy thirty three of the disc tracks, leaving thirty one for the programs, and the data files of the computer users. Considering that SIMUL8 alone requires eight of these thirty one tracks, it is obvious that the available storage was inadequate for the needs of the computer users.

Two alternative means of providing bulk storage for the computer were investigated. They were

. to extend the capacity of the present disc storage system, or

. to incorporate a magnetic tape storage system as a "back up" to the disc.

Magnetic tape was found to be the best choice for a number of reasons.

. Magnetic tape storage systems are less expensive than disc systems.

. The storage capacity can be increased to any desired level by purchasing more tape, which is very reasonably priced.

. Magnetic tape storage allows programs and data that are not being used to be completely removed from the computing system, leaving the disc free for the temporary storage of the programs of other users.

Because the magnetic tape devices, which were available for the EAI 640 computer, were prohibitively priced, a commercially available magnetic tape transport (DEC TU55) was purchased, and a suitable controller to interface this unit to the EAI 640 computer was designed and built within the department.

8.2 Design Constraints

The DECTAPE TU55 tape transport, as supplied by the manufacturer, consisted of the following:

. tape drive motors and brakes, and the associated circuitry to control tape motion, direction and speed.

. tape guides and spools to accommodate standard three quarter inch tape.

. a ten channel magnetic head, which is used for both reading and writing data. These ten channels are paired to provide five information channels. The reliability of the data is increased by this information redundancy.

. switch controls to allow manual manipulation of the tapes, and to provide a write-lockout for the protection of tape contents against accidental erasure.

A summary of the characteristics of the tape transport is given in table 8.1.

Four channels of communication are available on the EAI 640 computer, allowing the transfer of data and control information between the C.P.U. and the peripheral devices. The functions of these four channels are

. to transfer data words from the C.P.U. to a specified peripheral device.

. to transfer data words from a specified peripheral device to the C.P.U.

. to transfer control words from the C.P.U. to the peripheral device, to control the function of the peripheral.

. to transfer control words, which describe the current state of the peripheral, from the peripheral device to the C.P.U.

Each information channel carries sixteen information bits and a parity bit, and is accessed by executing one of four computer instructions.

Table 8.1 Characteristics of DEC TU55 tape transport.

Tape speed	93 ± 12 in. sec^{-1}
Stop time	150 msec
Start time	150 msec
Turn around time	250 msec
Tape width	0.75 in
No of tracks	10
No of accessible channels	5 (each channel contains 2 paired tracks to increase data reliability)
Tape length	3100 in
Tape capacity	2.4×10^6 bits $reel^{-1}$
Bit density on each track	350 ± 55 bits in^{-1}

8.3 System Principles

Each reel of magnetic tape supplied with the TU55 transport has a capacity to store one third of the disc contents. As a result the disc was arbitrarily divided into three regions of nearly equal size. Two of these regions are set aside for the storage of the system programs, and the third is used as a "scratch" area, in which an operator's programs can temporarily reside when they are being used. The magnetic tapes are then used to store an image of the scratch or "user" area, as it has been named, each operator maintaining a copy of his own user area. The operator's copy of the user area is then loaded onto disc before use, where it may be updated with standard disc operating system procedures. The updated user area is then copied onto magnetic tape at the end of a computing session, leaving the disc free for other operators, while maintaining an up to date version of that operator's programs.

The transfer of disc images between magnetic tape and disc is handled by a utility program (MUTIL, cf. section 8.8), which is simple to operate and allows the transfers to be completed quickly and conveniently.

Hardware to interface the tape transport to the computer can be relatively simple and still allow the above operating philosophy to be achieved because

- random access to information on magnetic tape is not required, and
- compatibility with other magnetic tape systems is not required.

So that full compatibility with the DECTAPE system could be achieved at a later date with a minimum of inconvenience, the structure of that system was adhered to wherever possible.

8.4 Tape Format

The ten information tracks in the TU55 tape transport are paired to increase data reliability, thus providing five independent information channels. These channels are allotted as follows:

- three channels for data storage,
- one channel to synchronize the position of the data bits, called the TIMING channel or BIT CLOCK,
- one channel to indicate the position of the six lines of data which form a particular sixteen bit word. This channel, which is called the MARKING channel, also indicates the position of the physical start and end regions of tape.

The organization of the data bits and timing information is shown in figure 8.2. Each frame, as a group of six lines on the tape is called, provides storage for eighteen bits of information. Of these, seventeen are used for storing the sixteen bit data word and its associated parity bit. The eighteenth bit defines whether the contents of that frame contains valid data. The function of this bit is described in section 8.5.

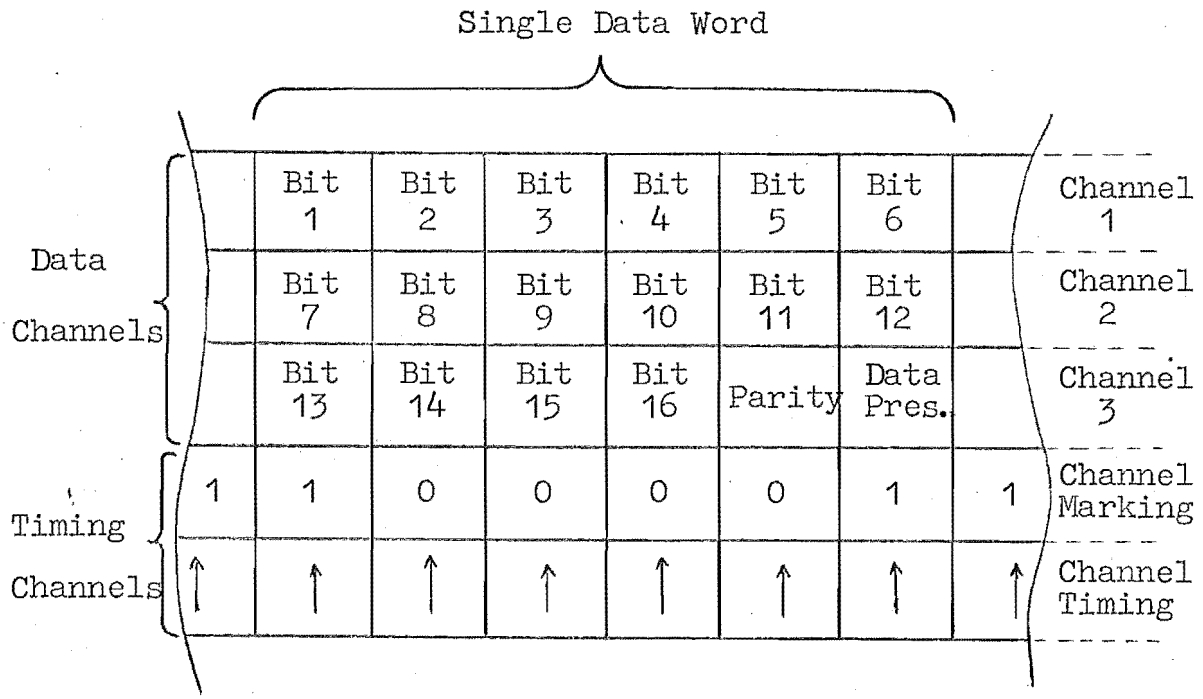


Figure 8.2 Layout of data and timing bits in one word on magnetic tape.

Special binary codes stored on the MARKING channel inform the controller when the physical ends of the tape are approached. On detecting the "start", or "end" codes, the controller automatically stops, or rewinds the tape respectively.

Both the TIMING and MARKING channels are prerecorded on the tape before reading or writing of data occurs. They are used in controlling the transfer of data between the tape and tape controller on all read and write operations.

8.5 Magnetic Tape Controller Hardware - Data Transfer

A block diagram indicating the main features of the magnetic tape controller is shown in figure 8.3. Data words are transferred between the tape controller and the computer in parallel, whereas, transfers between the tape controller and the tape are serial for each of three parallel data channels. The parallel to serial, and serial to parallel, conversions inherent when writing, and reading data, respectively, are effected by three six bit shift registers (cf. figure 8.3). All data words transferred between the computer and the controller are first stored in intermediate buffer registers. This allows the transfer to occur on a "no wait" basis. For example, when data is being transferred from computer to controller, a new word may be transmitted by the computer at any time following the transfer of the previous word from the input buffer register to the shift registers. Similarly when reading data from tape a word can be transferred to the computer any time after it is moved into the output buffer register, provided this is done before the next word is completely recovered.

In the standard DECTAPE system it is necessary, once writing has commenced, for the computer to maintain the supply of data words to the tape controller throughout the writing of an entire data block. Because the tape system described here is not arranged to write on independent blocks, but rather, writes continuously over the tape length, an alternative procedure is required, to define which regions of the tape contain data. Such a procedure may be implemented using the concept of a DATA PRESENT BIT which was

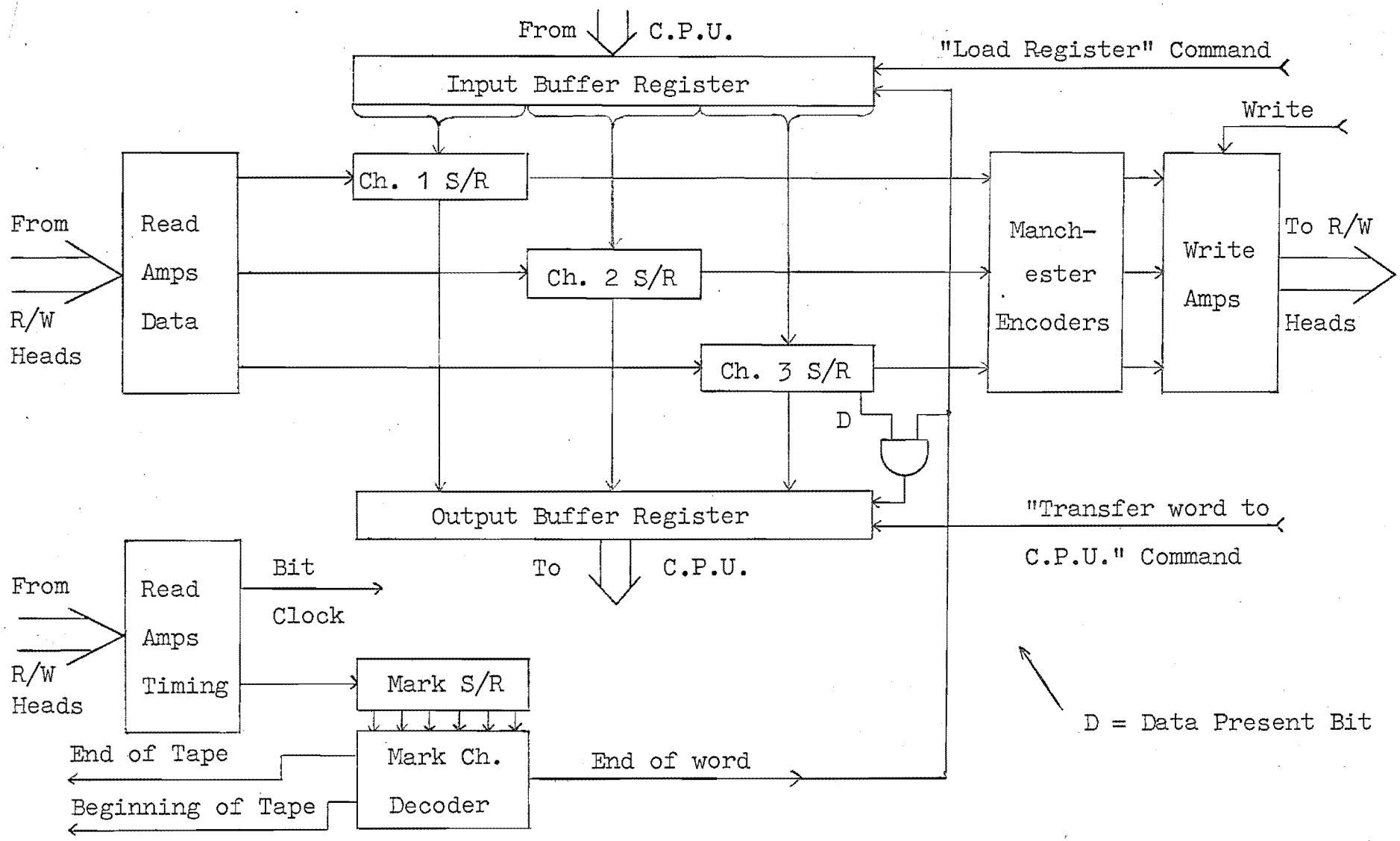


Figure 8.3 Main features of the data transfer sections of the controller.

introduced in section 8.4 Each word space on the tape contains one bit which defines whether the contents of the word are data, or not. This bit is set if the word of which it is a constituent, originated in the computer. Otherwise it is reset. When the tape is subsequently read, only those words which have their DATA PRESENT BIT set are recognised by the controller and transferred to the computer. All other words are ignored. If for some reason the computer is busy with other peripheral devices when a data word is required by the tape controller, a dummy word is written onto tape. This word is ignored on subsequent read operations. The use of the DATA PRESENT BIT as a record separator is described later (cf. section 8.7).

8.6 Magnetic Tape Controller Hardware - Control

The state of the magnetic tape controller (e.g. whether it is reading, writing etc.) is controlled directly by the computer and may be changed at any time by a control word which is transferred from the computer to the controller. This word contains the desired states of read, write and tape motion in the controller.

The current state of the controller may be tested by transferring a status word from the controller to the computer. This word describes the current state of read/write and tape motion in the controller, and also defines any error conditions which have occurred in the controller. Error flags are set when writing is attempted with the write-lock switch on, or, when a word read from the tape is not transferred to the computer before it is overlaid by the next word read.

The prerecording of the TIMING and MARKING channels is timed by special circuitry in the controller. However, the codes written on the marking channel, which define the various regions of the tape, are generated within the computer, so that the complexity of this hardware is minimized.

8.7 Data Format on Tape

Data stored on magnetic tape is a direct copy, or image, of the disc contents. For this reason, the format of the data on tape is constrained by the storage structure of the disc. The information on the disc is arranged in blocks corresponding to a physical track of the disc, each track holding 5632 words. To allow time for the transfer of a single track of information between disc and computer, it is necessary to have short gaps between data blocks on the tape. These gaps are implemented using the DATA PRESENT BIT, which was described in section 8.5. By not transferring information to the tape during disc-to-computer transfers, this bit is automatically reset by the tape controller, for all words written during this time, leaving a gap between records on the tape. When the tape is being read back to disc, these gaps allow time for the transfer of information from the computer to the disc. The layout of the information and the interrecord gaps on the tape is shown in figure 8.4.

The first word of each block contains the number of the disc track from which the subsequent information was taken. Following this are the 5632 data words, and then a longitudinal parity or checksum word, formed by taking the exclusive OR of the 5632 data words.

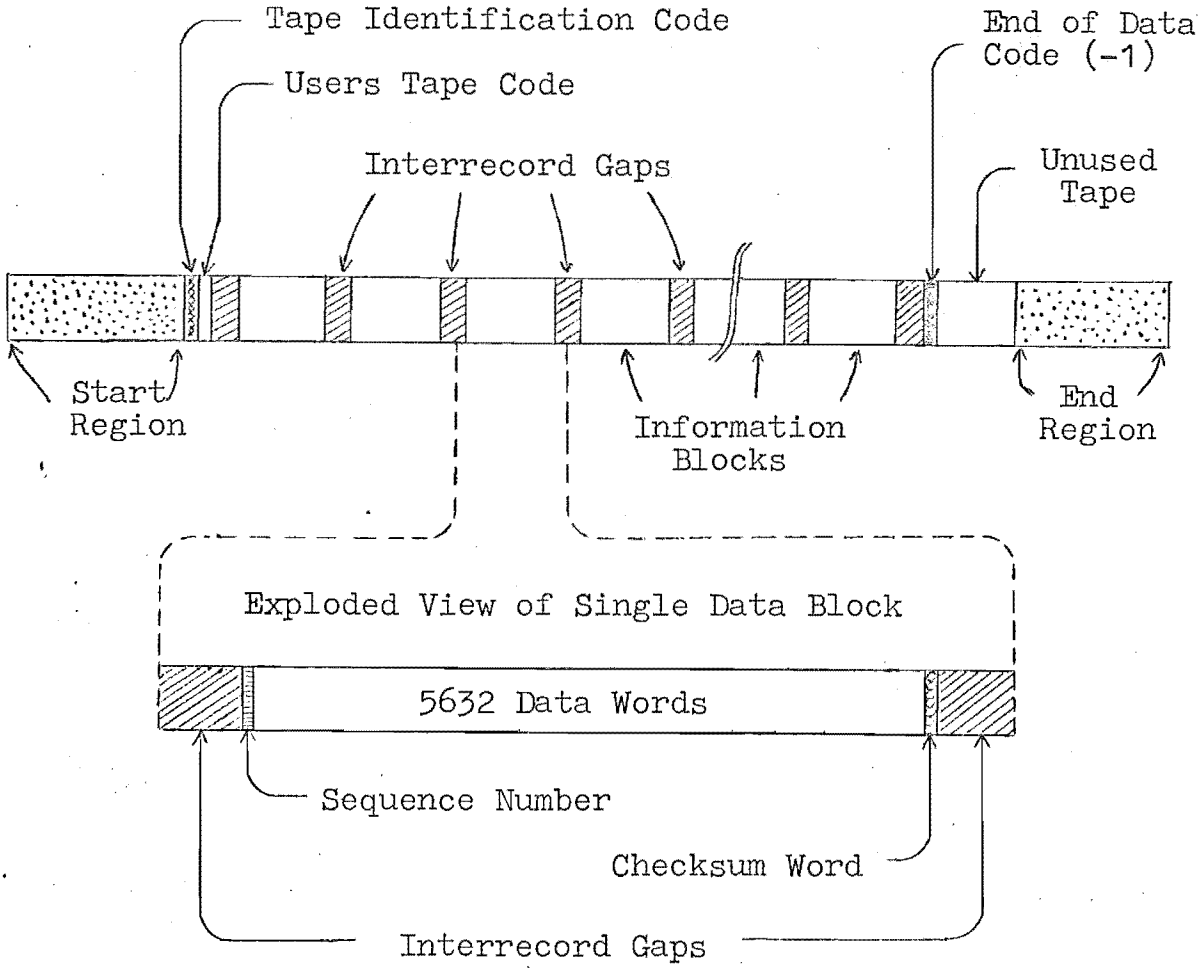


Figure 8.4 Layout of data blocks and interrecord gaps on magnetic tape.

The checksum word, when used in conjunction with the parity bits from each word, provides a means of detecting errors that occur when a tape is read. Providing only one error occurs in any given data block, the checksum and parity information may be used to correct this error.

At the start of a tape, user and tape identification codes are stored. The user identification code is provided to protect a tape against accidental erasure. This code must be identified by the user before any writing on tape can proceed.

The user and tape identification fields are followed by the information blocks, each separated by interrecord gaps which are about 400 words in length. Following the last interrecord gap, a track number of -1 signifies the end of the information on the tape.

The software which controls the tape, and places the information on it in the manner described above, is discussed in the next section.

8.8 Magnetic Tape Utility Program

All transfers of information between disc and magnetic tape, and the housekeeping of tapes, are handled by a utility program named MUTIL. This program which is normally resident on disc, is loaded into core using the disc operating system. On execution, MUTIL requests the name of the utility required. When the requested task is completed, the user is again asked for a utility name.

The function and names of some of the more commonly used utilities are listed below.

- DUMP - copies the contents of the user area of the disc onto magnetic tape, and then checks that information by reading it from tape and comparing it word for word with the disc contents.
- LOAD - reads and checks the information on magnetic tape and transfers it to disc, providing no unrecoverable errors are detected.
- FORMAT - writes the bit timing clock and the marking channel codes onto a new tape.

. USERID - writes the user and tape identification codes on a new tape.

. TERM - used when operations on the magnetic tape have been completed.

To summarize the operation of MUTIL, an example of its use is given. On commencing a computing period, the user loads and executes the program MUTIL. After placing his tape spool on the tape deck he requests the LOAD utility. When the tape contents are loaded onto disc, TERM is executed and the user may continue with this computer operations. When the computing session is completed, MUTIL is again loaded and the utility DUMP is requested. The disc contents are then written onto magnetic tape and verified. The tape spool may then be removed and stored for later use, the disc storage being left free for other users.

8.9 Conclusions

The magnetic tape system described in this chapter has provided much needed bulk storage for the EAI 640 digital computer in the Electrical Engineering Department. Computer users have adapted readily to using magnetic tape, and have found it both simple to use and reliable. Currently more than fifty reels of magnetic tape are in regular use.

Although the magnetic tape does not have the flexibility of a commercially available unit, it is more than adequate for the task in hand.

The major advantage of the system, as implemented, is the relatively low cost compared to commercial units, which were well outside the budget available to us.

The magnetic tape has allowed the program SIMUL8 (see chapter 7.0) to be developed far beyond what was at first thought possible. The operating system which provides the major part of SIMUL8's flexibility would not have been possible with the limited disc storage available before the magnetic tape system was commissioned.

Complete technical details of the magnetic tape controller, including full circuit and wiring diagrams, and detailed notes on operation and maintenance of the unit are lodged in the Hybrid Computer Laboratory, University of Canterbury, Christchurch, New Zealand.

CHAPTER 9

PARAMETER VARIANCE CONSIDERATIONS WHEN SIMULATING LINEAR SYSTEMS

The conventional simulation techniques, such as those discussed in chapters 2 and 7, provide a means of solving a number of differential equations which describe a model, for a particular set of model parameters. These parameters are chosen in such a way that they are typical for the process or system which is being modelled. However, because the parameters are random variables, which are different for each member of a species of a biological system, the simulation results describe the behaviour of only one member of the species. No information on the spread of the results is gained in this way. To obtain this it is necessary to carry out more sophisticated simulation procedures.

"Between animal" variations are incorporated into purely stochastic models by Lucas (1964), but there do not appear to have been any attempts to study continuous models in this way.

In this chapter a number of methods by which parameter statistics may be incorporated into the simulation of linear systems are described. The chief problem is how to obtain the required information with an economical amount of computing. A new method, called the "variant function technique", is introduced and is developed to the point where its potential advantages are apparent.

9.1 The Solution Ranges in a Linear System

The range of the solution of a set of differential equations with randomly distributed parameters is defined here as the upper and lower bounds of the solution, at a particular instant, for any permissible combination of parameter values. The range curves of a solution are defined to be the loci of the upper and lower bounds of the solutions over all times. The range curves for systems whose model parameters are unbounded are consequently meaningless, e.g. the solution range for normally distributed parameters is indefinitely large. However, the parameters of real practical systems are bounded. The two following observations may be made about the range curves.

I The range curves of a solution to a set of differential equations are, in general, not themselves solutions to the differential equations.

II Even if the upper and lower range curves are solutions to the differential equations, they do not always correspond to solutions of equations with extreme parameter values.

These two points are best demonstrated by a graphical example. We consider the linear system which is defined by the equations

$$\frac{dy_1(t)}{dt} = -k_1 y_1(t) \quad , \tag{9.1}$$

$$\frac{dy_2(t)}{dt} = k_1 y_1(t) - k_2 y_2(t) \quad , \tag{9.2}$$

where k_1 may exist in the range

$$0.1 < k_1 < 0.2 \quad , \tag{9.3}$$

$$k_2 = .05 \quad , \tag{9.4}$$

and the initial values of y_1 and y_2 are

$$y_1(0) = 1 \; ; \; y_2(0) = 0 \; . \tag{9.5}$$

Figure 9.1 shows three solutions for $y_2(t)$ where k_1 takes on values of 0.1, 0.15 and 0.2 respectively. The estimated positions of the range curves are also shown in figure 9.1. Note the discontinuity in the slope of $y_2(t)$ at point A in the figure. This emphasises observation I, because it shows that the range curve is not a solution to equations (9.1) and (9.2). At point B in the figure, an intermediate value of k_1 (viz. $k_1 = .15$) touches the upper range curve, which verifies observation II.

With a system of equations which contains a large number of randomly distributed parameters, many values of these parameters would have to be used in conventional simulations to obtain a reasonable estimate of the position of the range curves. While such a procedure is practicable, it is doubtful whether the information it yields is valuable. Even if the parameters are rectangularly distributed (i.e. their values occur with equal likelihood over a finite range), the solutions are not rectangularly distributed at each instant. This is exemplified by the lengths of the regions "a" and "b" in figure 9.1. If the solutions were rectangularly distributed these distances would be equal.

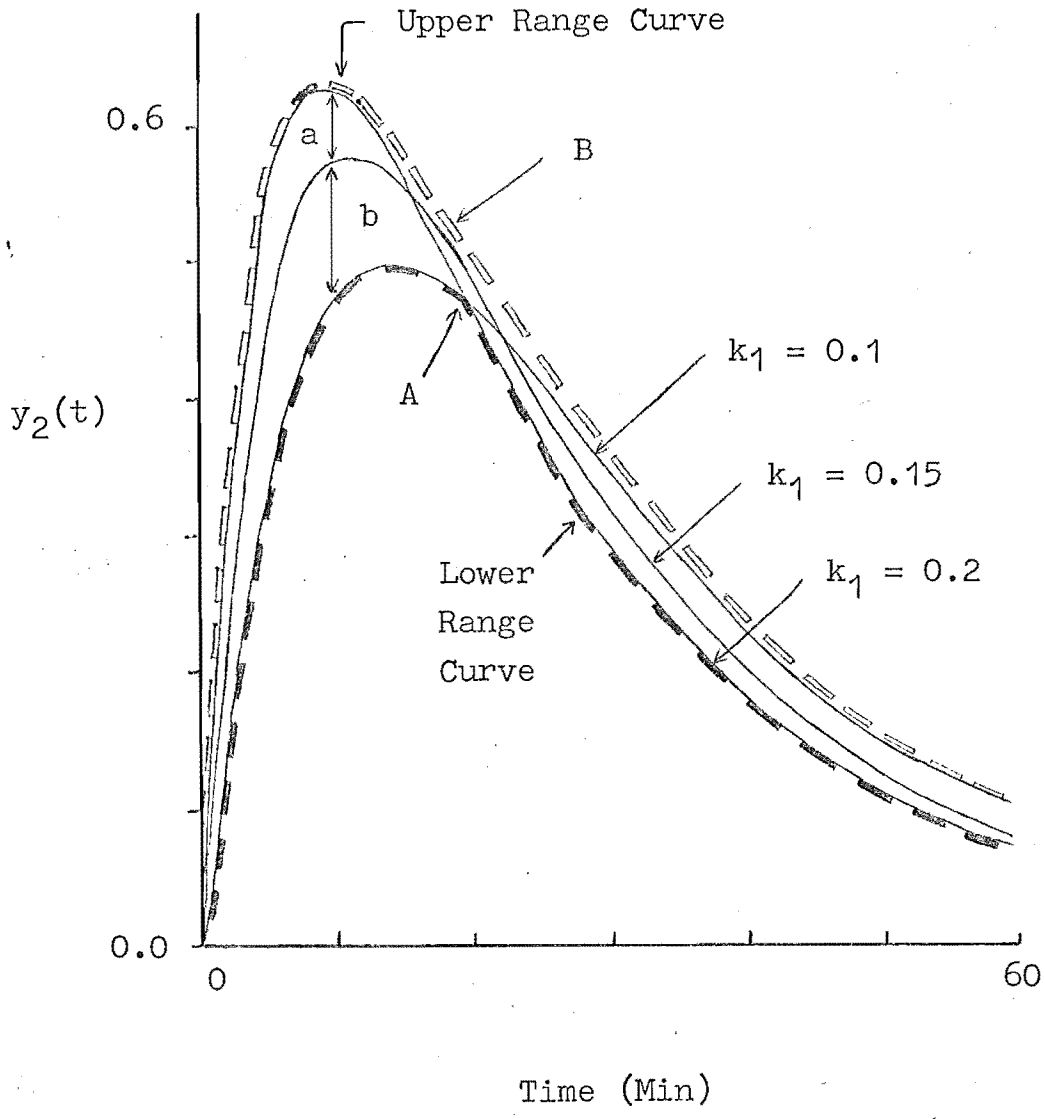


Figure 9.1 Solutions for y_2 in equations (9.1) and (9.2) for three values of the parameter k_1, and estimated positions of the lower and upper range curves. Note the discontinuity at point A in figure.

While the range curves are easy to define and (in principle) to compute, we see that they do not provide a clear indication of the spread, or distribution, of the solutions. We need more sophisticated curves to display this. It is likely that curves representing the mean and standard deviations of the solutions would be more meaningful.

9.2 Mean and Standard Deviation Curves in a Linear System

There are a number of methods by which the statistical form of the solutions of a set of linear differential equations with randomly distributed parameters may be investigated. All involve large amounts of computation. Two well known methods are now described.

We consider the set of linear differential equations

$$\frac{d}{dt} y(t) + A\, y(t) = f(t) \quad , \tag{9.6}$$

where $y(t)$ is the state variable vector, A is a matrix of randomly distributed system parameters and $f(t)$ is the system forcing function.

By using the state transition matrix e^{-At}, which is defined as

$$e^{-At} = \sum_{i=0}^{\infty} \frac{1}{i!} (-At)^i \quad , \tag{9.7}$$

we find that the solution of equation (9.6) is (see Appendix 3)

$$y(t) = e^{-At}\, y(0) + \int_0^t e^{-A(t-\tau)}\, f(\tau)\, d\tau \, . \tag{9.8}$$

In order to show how the amount of computation may easily become unacceptably large, we consider the natural response of equation (9.6), corresponding to the solution of

$$\frac{d}{dt} y(t) + A\, y(t) = 0 \quad , \tag{9.9}$$

which is (see Appendix 3)

$$y(t) = e^{-At}\, y(0). \tag{9.10}$$

Two alternative methods of solving equation (9.9) are now described. The first involves the series expansion of equation (9.10), while the second involves solving equation (9.9) many times with randomly chosen values for the A matrix.

9.3 Series Expansion of the Transition Matrix

Let the mean of a statistical quantity be denoted by a superbar, e.g. the mean of A is $\overline{A}$. Using equation (9.7), the mean of equation (9.10) may be written as

$$\overline{y(t)} = \overline{\sum_{i=0}^{\infty} \frac{1}{i!} (-At)^i\, y_0} \; . \tag{9.11}$$

This is clearly different to the solution $\hat{y}(t)$ to equation (9.10) which is got by using the mean value of A, i.e.

$$\hat{y}(t) = \sum_{i=0}^{\infty} \frac{1}{i!} (-\overline{A}t)^i\, y_0 \; . \tag{9.12}$$

Comparison of equations (9.11) and (9.12) shows that the mean solution of a set of differential equations with

randomly distributed parameters is, in general, different from the solution to those equations for the mean values of the parameters. Furthermore, the mean solution is itself not necessarily a solution of those differential equations.

Equation (9.11) may be rewritten as

$$\overline{y(t)} = \sum_{i=0}^{\infty} \overline{A^i} \, (-t)^i \, \frac{y_0}{i!} \, . \qquad (9.13)$$

For the following two reasons, equation (9.13) can be difficult to compute.

1. If At is large, many terms of the power series are needed to provide good accuracy. Note that in practice, because A is a matrix, the size of At might be defined by $\|A\| t$, where $\|A\|$ is the norm of the matrix A (cf. Liou, 1966).
2. Calculation of $\overline{A^i}$ requires the high order moments of the elements of A, for example moments of the form $\overline{A_{ij}^m \, A_{k1}^n}$. In general these moments are not known and estimates of these must be made.

If the original equations can be transformed to a set of equations in which the elements of A are statistically independent of one another, the need to estimate the cross product moments (for example the covariance) can be eliminated. Even in this situation moments of the form $\overline{A_{ij}^m}$ are required, for large m.

In the models of interest, At is generally large, and the elements of the matrix A are highly dependent on each other (for example, if equations (9.1) and (9.2) are transformed to the form of equation (9.9) then $A_{11} = - A_{21}$).

Thus, for the two reasons given above, this analysis technique is impracticable. To gain estimates of the variance, the above two reasons have an even more dramatic effect on the computation.

9.4 Multiple Simulation Techniques

A further method of calculating the statistical form of the solutions y(t) in equation (9.9) involves solving the equations many times using different values for the elements of the A matrix for each run. The many solutions are then analysed at a number of discrete time intervals to determine, for example, the mean and standard deviation curves. The values for the elements of the A matrix are chosen so as to reflect the statistical properties of those parameters.

There are two possible ways of choosing values for the elements of the A matrix. First, they may be chosen systematically, say ten values for each parameter, and then each combination of the ten values used in separate simulations. This would require 10^N simulations, where N is the number of parameters. It is obvious that this method of selecting parameters is practicable only for small models.

The second way of choosing values for the elements of the A matrix is to select them at random from populations which have the same statistics as the parent populations. This is similar to the "Monte Carlo" method used by Schull and Levin (1964). We note, however, that in their simulations, model variables rather than model parameters are the random variables.

Computationally, Monte Carlo parameter selection provides a significantly faster method of investigating the statistical form of the solutions to equation (9.9), compared with systematic selection of parameter values.

Monte Carlo selection has a further advantage. The estimates of the mean curve, for example, are asymptotic to the true mean curve, and the accuracy of the estimates increase with the number of simulations. After only a few simulations, a crude estimate of this curve is obtained. By studying the changes in the curve, the degree of numerical convergence can be estimated, and the simulations stopped when the results appear to be sufficiently accurate.

9.5 The Variant Function Technique

A third method of investigating the statistical form of the solutions to equation (9.6) is now developed. The elements of the matrix A are varied in such a way that the statistical distribution of the solutions is estimated from a single simulation.

We consider the equation

$$\dot{\eta}(t) + \dot{\Omega}(t)\,\eta(t) = f(t) \quad , \tag{9.14}$$

which has a solution (see Appendix 3).

$$\eta(t) = e^{-\Omega(t)}\left[\int_0^t e^{\Omega(\tau)}\, f(\tau)\, d\tau + \eta(0)\right] \quad . \tag{9.15}$$

If the time varying matrix $\Omega(t)/t$ has the same statistical properties, measured over a long time, as the constant matrix A then $\eta(t)$ will have the same statistical properties

as y(t) in equation (9.6). Statistical properties are defined by the moments of the distributions. For the above condition to be met, all moments of $\Phi(t)/t$ and A must be the same. So, if i, j and m are positive integers, we require that

$$\overline{(\Phi_{ij}(t)/t)^m} = \overline{(A_{ij})^m} \qquad (9.16)$$

where the subscripts i, j denote the element of Φ or A in the ith row and the jth column. Also, if the elements of A are not independent, then the elements of $\Phi(t)/t$ must have the same interdependence.

The function $\Phi(t)/t$ is hereinafter called the "variant function", denoted by the symbol V(t). Thus

$$V(t) = \Phi(t)/t \; . \qquad (9.17)$$

To show how equation (9.16) may be satisfied, we consider a single element $V_{ij}(t)$ of the matrix function V(t). Because the derivative of $t\,V_{ij}(t)$ with respect to time (viz. $\Phi_{ij}(t)$) is required in equation (9.14), it is essential that $V_{ij}(t)$ be at least piecewise continuous in time. The simplest way that $V_{ij}(t)$ may satisfy equation (9.16), and be piecewise continuous in time, is for it to be a periodic function of period T_{ij}. Thus

$$V_{ij}(t) = V_{ij}(t + mT_{ij}); \quad m = 1, 2 \ldots \qquad (9.18)$$

The form of $V_{ij}(t)$ over one interval T_{ij} is then directly related to the probability distribution function $P(A_{ij})$, which is defined as the integral of the probability density (viz. $p(A_{ij})$; cf. Bennett, 1956) of A_{ij}, i.e.

$$P(A_{ij}) = \int_{-\infty}^{A_{ij}} p(\alpha)\, d\alpha . \tag{9.19}$$

If the functional relationship describing $P(A_{ij})$ is represented in the piecewise linear form

$$P(A_{ij}) = \frac{2t - mT_{ij}}{T_{ij}} , \quad mT_{ij} < t < (m + \tfrac{1}{2})\, T_{ij}$$

$$= \frac{(m + 1)T_{ij} - 2t}{T_{ij}} , \quad (m + \tfrac{1}{2})T_{ij} < t < (m + 1)T_{ij} , \tag{9.20}$$

and if

$$V_{ij}(t) = A_{ij}(P) , \tag{9.21}$$

then equation (9.16) is satisfied. The above result is shown graphically in figure 9.2. Combining equations (9.20) and (9.21) yields

$$V_{ij}(t) = A_{ij}\left(\frac{2t - mT_{ij}}{T_{ij}}\right), \quad mT_{ij} < t < (m + \tfrac{1}{2})T_{ij}$$

$$= A_{ij}\left(\frac{(m + 1)T_{ij} - 2t}{T_{ij}}\right), \quad (m + \tfrac{1}{2})T_{ij} < t < (m + 1)T_{ij} \tag{9.22}$$

The procedure outlined above provides a means of generating the variant function $V_{ij}(t)$ which corresponds to a single model parameter A_{ij}. However, a number of difficulties can arise:

I. if the probability distribution function $P(A_{ij})$ is asymptotic to the probability extremes 0 and 1, the variant function contains infinite values in two regions of every period T_{ij}. In these situations the probability distribution functions

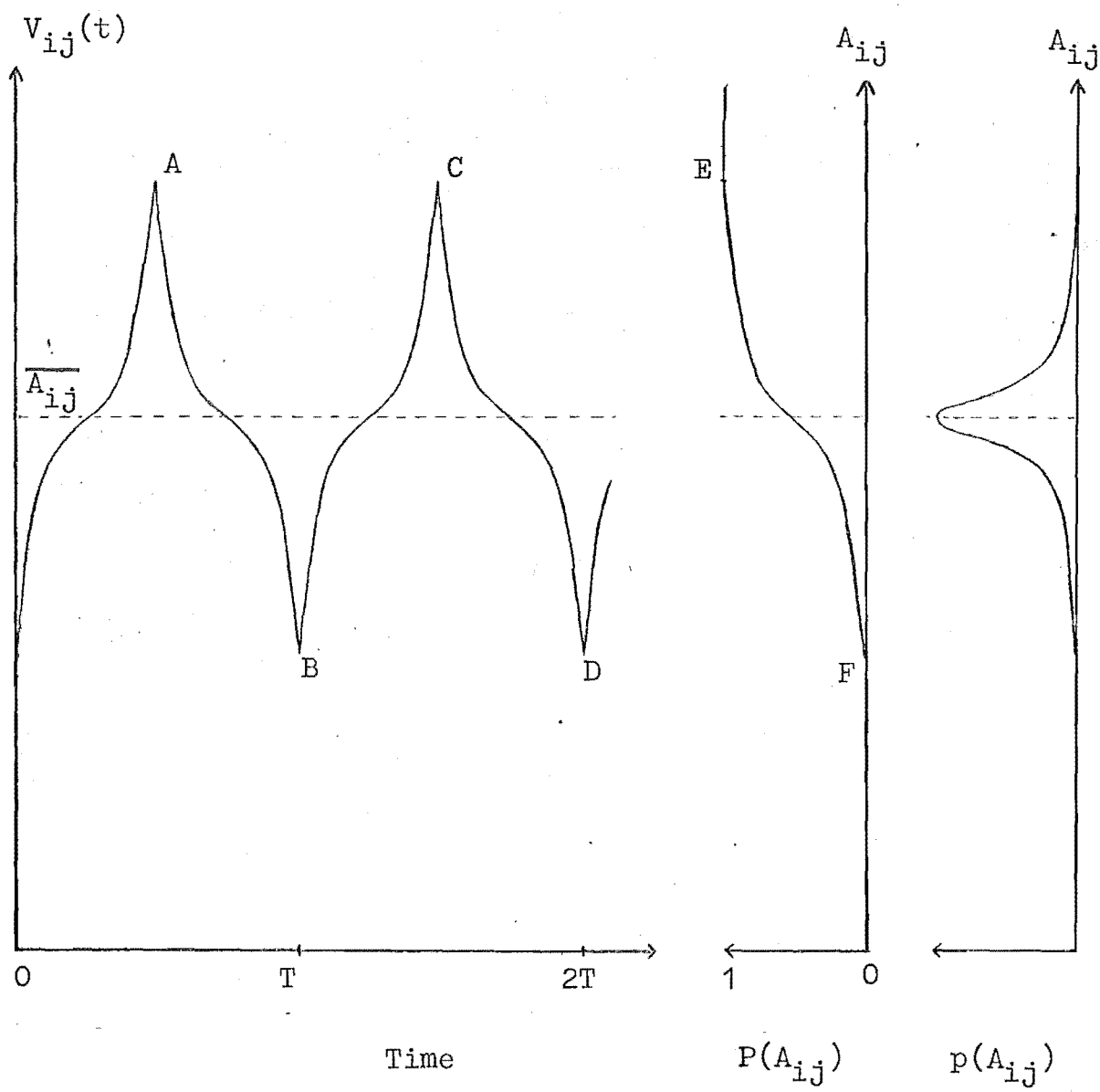


Figure 9.2 An example of a variant function showing its relationship to the probability distribution function. Note that segments A-B, C-D and E-F are identical, and that A-B and B-C are mirror images. The function $V_{ij}(t)$ has the probability density shown at the right. Note also that

$$P(A_{ij}) = \int_0^{A_{ij}} p(\alpha)d\alpha .$$

must be truncated at convenient points.

II some probability distribution functions are difficult, if not impossible to invert to the form $A_{ij}(P)$. In these cases numerical approximations to the inverse functions must be formed by using graphical techniques.

If the model parameters are independent of each other then the variant functions generated for each model parameter must also be independent. This is assured if the variant functions are orthogonal. If $V_{ij}(t)$ and $V_{kl}(t)$ are orthogonal then

$$\int_0^\infty (V_{ij}(t) - \overline{V_{ij}})(V_{kl}(t) - \overline{V_{kl}})dt = 0 \qquad (9.23)$$

The orthogonality condition is met if the periods T_{ij} in equation (9.22) are different for each variant function.

9.6 The Variant Function Technique used in a Simple Example

The variant function technique is now applied to a small model with rectangularly distributed parameters. The model to be studied has the form shown in equation (9.9) with parameters

$$A = \begin{bmatrix} k_1 & 0 \\ -k_1 & k_2 \end{bmatrix}, \qquad (9.24)$$

and initial conditions

$$y(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \qquad (9.25)$$

The parameters k_1 and k_2 are assumed to have rectangular distributions of the form

$$p(k) = 50 \quad , \quad .09 < k < .11$$

$$= 0 \quad , \quad \text{otherwise,} \tag{9.26}$$

where

$$p(k) = p(k_1) = p(k_2) \; . \tag{9.27}$$

Using equation (9.22) the variant functions corresponding to the elements of A in equation (9.24) can be shown to be

$$V_{ii}(t) = .09 + \frac{1}{50\ T_{ii}} (2t - mT_{ii}) \quad ,$$

$$mT_{ii} < t < (m + \tfrac{1}{2})\ T_{ii}$$

$$= .09 + \frac{1}{50\ T_{ii}} ((m + 1)T_{ii} - 2t) \; ,$$

$$(m + \tfrac{1}{2})T_{ii} < t < (m + 1)T_{ii} \quad , \tag{9.28}$$

where the index i may take on the values 1 or 2,

$$V_{12}(t) = 0 \quad , \tag{9.29}$$

and

$$V_{21}(t) = - V_{11}(t) \; . \tag{9.30}$$

The two periods T_{11} and T_{22} are assigned values of

$$T_{11} = 1.59 \tag{9.31}$$

and

$$T_{22} = .556 \quad , \tag{9.32}$$

and the auxiliary equation (equation (9.14)) solved using SIMUL8 (cf. chapter 7.0) for the time period $0 < t < 50$. The solution for $\eta_2(t)$ is shown in figure 9.3.

Figure 9.3, while containing all the information required to define the statistical form of the solution to equation (9.9), is not easily interpreted visually. In the next section a method of improving the presentation is described.

9.7 Estimating the Means and Standard Deviations from the Solutions found by the Variant Function Technique

Direct methods of finding the mean and standard deviations of the solution $\eta_2(t)$ in figure 9.3 as functions of

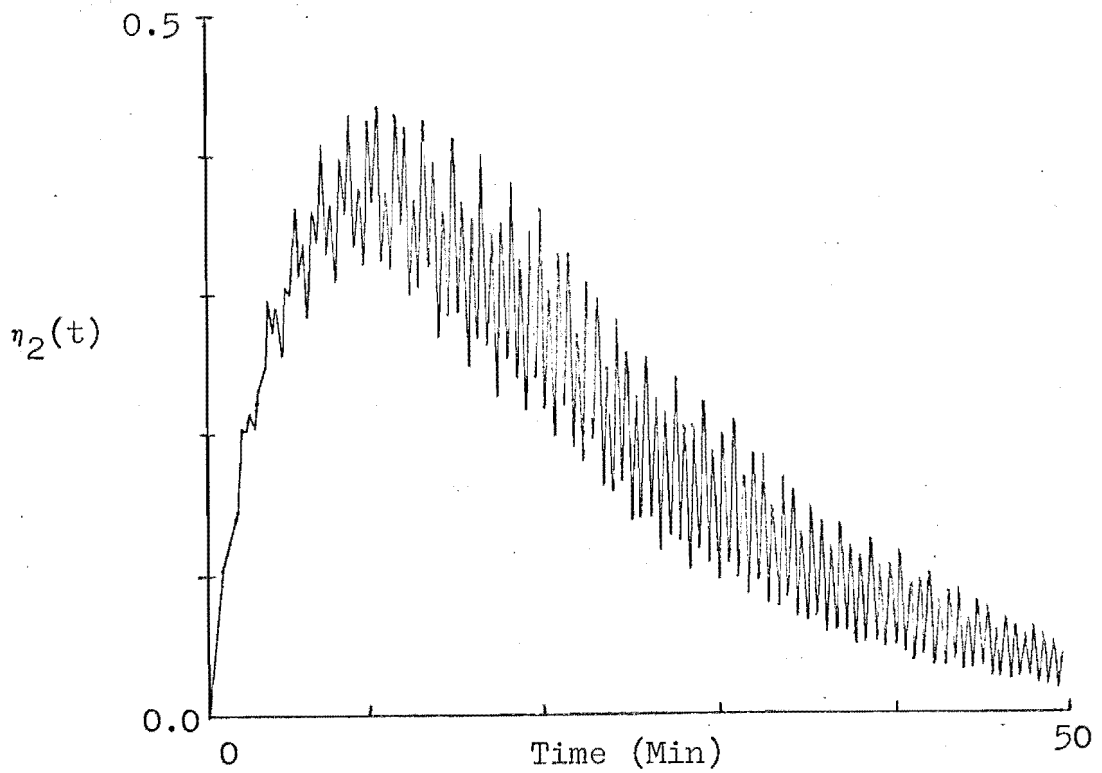


Figure 9.3 Solution for $\eta_2(t)$ using the variant function technique. The parameters of the system are uniformly distributed, so that their variant functions have linear sawtooth forms.

time are unsatisfactory. For example, if the solution is divided into a number of small time intervals of length Δt, the mean solution may be found by interpolating the mean values obtained for each time interval. But, if Δt is too small, the solution has an oscillatory form similar to that shown in figure 9.3. Conversely, if Δt is too large the mean solution deviates significantly from the true mean because the mean solution is itself a time varying function. For these reasons a compromise must be reached.

A more practical method of estimating the mean and standard deviation curves is provided by Otterman (1960) who proposes the use of certain estimates - termed "exponentially mapped past" (EMP) estimates - of the two statistics. These allow the mean and standard deviation to be found as functions of time, even when the function to be analysed is heavily contaminated with noise.

The EMP mean estimate $\bar{z}(t)$ of the variable $z(t)$ is defined by Otterman (1960) to be

$$\bar{z}(t) = \rho \int_{-\infty}^{t} z(\tau)\, e^{-\rho(t-\tau)}\, d\tau \quad , \tag{9.33}$$

which corresponds to the solution to the differential equation

$$\dot{\bar{z}}(t) = \rho(z(t) - \bar{z}(t)) \quad . \tag{9.34}$$

Otterman also defines the EMP variance $\sigma^2(t)$, where

$$\sigma^2(t) = \rho \int_{-\infty}^{t} (z(\tau) - \bar{z}(\tau))^2 e^{-(t-\tau)}\, d\tau \quad , \tag{9.35}$$

which corresponds to the solution to the equation

$$\frac{d}{dt}(\sigma^2) = \rho(z(t) - \bar{z}(t))^2 - \rho\sigma^2(t) \quad . \tag{9.36}$$

By solving equations (9.34) and (9.36) simultaneously with the equations of the model, the EMP mean and variance are found during the simulation procedure. The EMP mean is used directly to indicate the position of the mean solution, while the EMP variance is used to form a "variation band", which is defined here to be the area bounded by the curves representing, respectively, the sum and the difference of $\bar{z}(t)$ and $\sigma(t)$.

Using the model whose solution is shown in figure 9.3, the EMP mean and variance are calculated by solving equations (9.34) and (9.36), and the variation band is found. These results are shown in figure 9.4, which is more readily interpreted than figure 9.3.

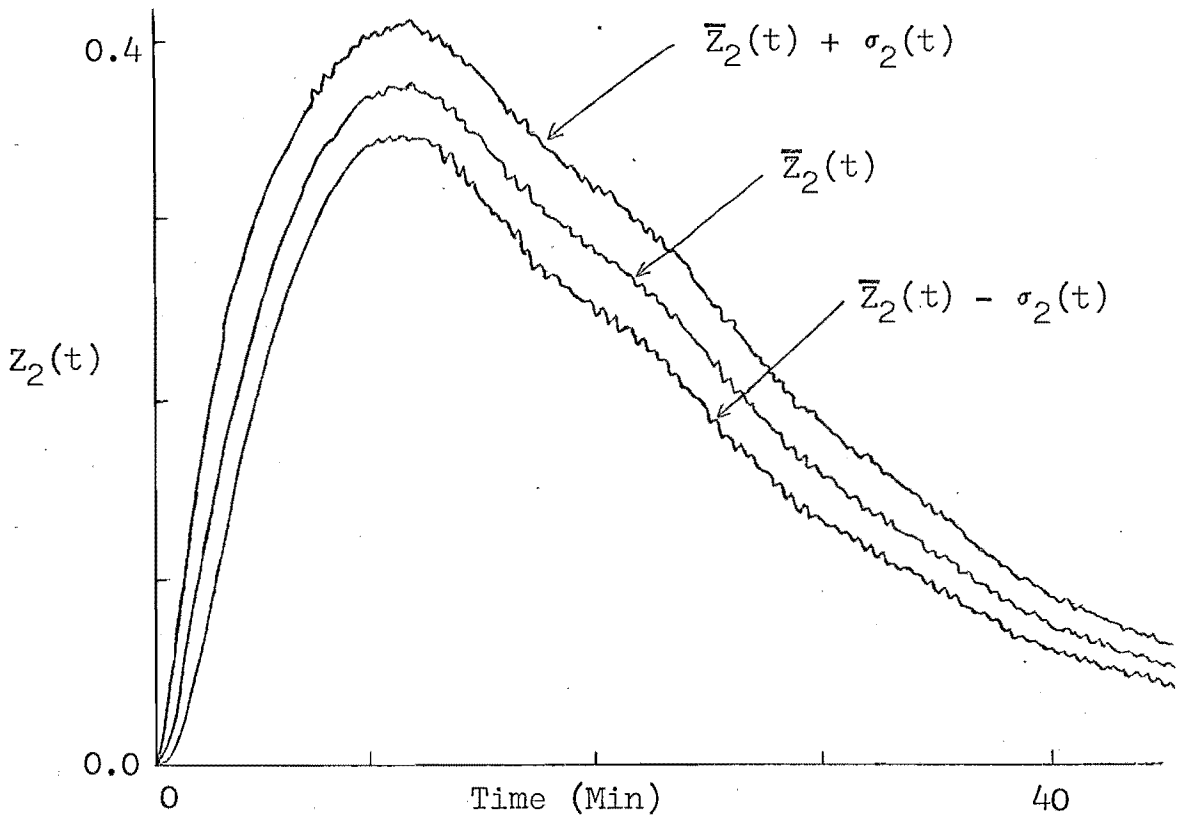


Figure 9.4 EMP mean and EMP variation band for the solution shown in figure 9.3.

9.8 Accuracy of the Variant Function Technique

Computational errors introduced during the simulation of systems using the variant function technique can become large if sufficient care is not taken.

The simulation program SIMUL8 (see chapter 7), in its standard form, incorporates a fast floating point arithmetic subroutine using a short mantissa (16 binary bits), in contrast to the standard EAI floating point subroutines which use a 24 bit mantissa. In the early stages of the investigation the short mantissa was found to have insufficient accuracy and the 24 bit mantissa had to be implemented at the expense of increased computation time. In fact the solution time was increased by a factor of two. In figure 9.5, the EMP mean curves, found using 16 and 24 bit mantissa's are compared. The solution using a 16 bit mantissa has significantly more distortion than would be expected to occur in the solution of a second order system.

A further source of error is the truncation introduced by the integration algorithm. In SIMUL8, a second order predictor formula is used. With larger systems, where the number of variant functions used is also large, it may be necessary to use a higher order integration algorithm, or a predictor-corrector procedure such as Milnes method (cf. Noble, 1964), so that errors introduced by the integration procedure can be kept to an acceptable level.

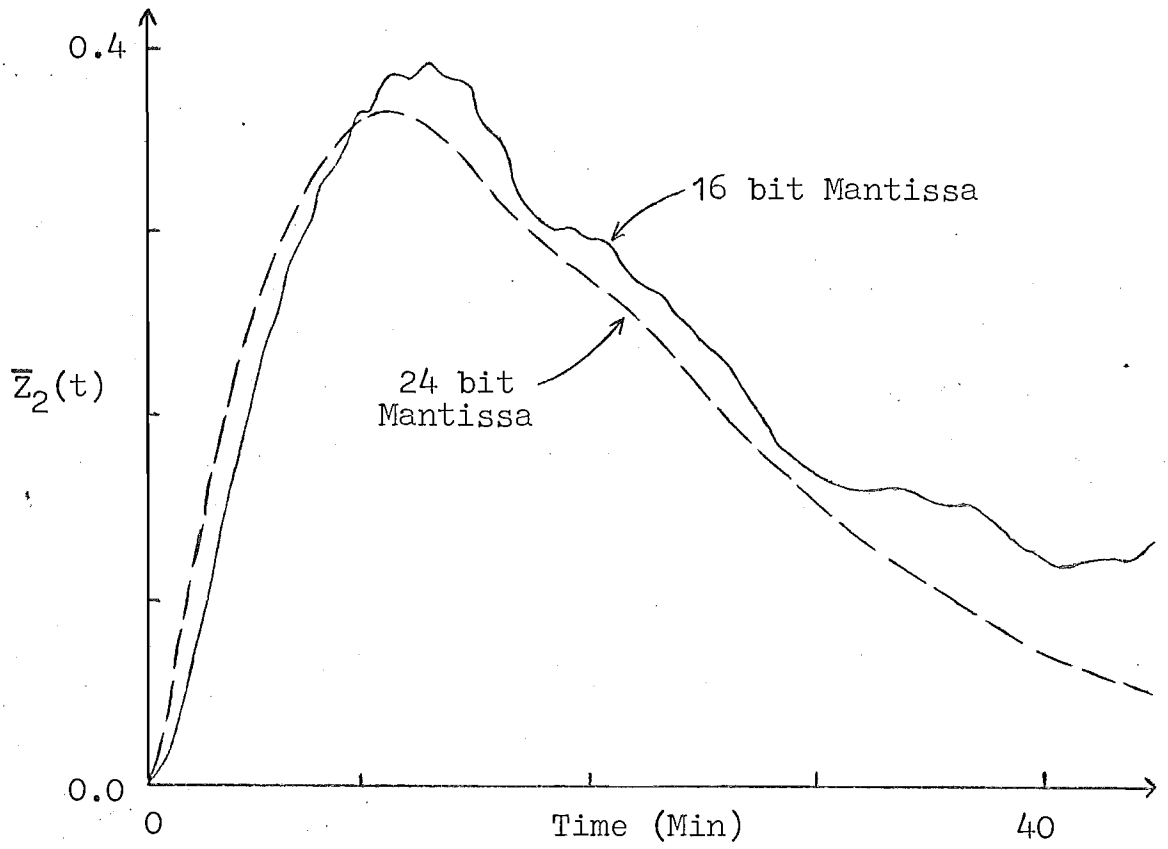


Figure 9.5 EMP mean solutions found with two different lengthed mantissas in the arithmetic procedures. Note the deviations of the solution when the shorter mantissa is used.

9.9 Summary

In this chapter, three methods of analysing linear differential equations whose parameters are randomly distributed are described. All three provide computationally sound procedures for determining the statistical spread of the solutions to a set of linear differential equations.

Method 1 involves the series expansion of the state transition matrix e^{-At} (cf. section 9.3). If the magnitude

of At becomes large, many terms of the series are required for satisfactory convergence. Each successive term of this series contains progressively higher order statistical moments of the parameter distributions. Because these are usually unknown, suitable approximations to them must be calculated for each term in the series.

In Method 2, the system to be investigated is solved for many combinations of parameter values, selected on a random basis in such a way that they reflect the actual parameter distributions (cf. section 9.4). This technique has the advantage that the solutions usually converge onto the true solutions as more combinations of parameters are tried. The procedure can be terminated when the solution appears to have been determined to any desired accuracy. This second technique is ideally suited to implementation on a hybrid computer. The system equations are patched onto the analogue computer, allowing rapid generation of solutions. The digital computer may then modify the model parameters in a Monte Carlo fashion, and take samples from the solutions to form the mean curve and variation band. Hybrid computation offers significant advantages in computation time compared with pure digital computation, particularly when the system under investigation is large.

Method 3, the variant function technique (cf. section 9.5), uses the statistical distributions of the model parameters directly. These distributions are transformed to time varying functions, which are used as the "parameters" of an analogous set of differential equations. The solutions

to these equations have statistical properties which are directly related to those of the solutions to the original equations, and the mean solutions and variation bands can be extracted relatively easily.

Only one set of equations requires solution when using the variant function technique, and these equations are solved once only. For this reason, standard simulation procedures (e.g. SIMUL8, see chapter 7) can be used to implement the technique, and conversely, the technique can be incorporated into existing simulation studies to assess the affect of parameter variations on the model solutions.

The amount of computation required by the three techniques is large, and depends on the properties of the system being analysed. Computation time increases with the number of equations and parameters, with the duration of the simulation and with the reciprocals of the model time constants.

In general method 1 is impractical for all but a few cases. Only those systems which require relatively few terms in the series expansion can be analysed, and such systems would be better studied using one of the other two methods.

Method 2 is potentially fast, when hybrid computation is available, for solving large systems with relatively few parameters. When the digital computer is used the Monte Carlo technique has advantages for small systems but, as the number of parameters becomes large, the variant function

technique is more efficient computationally. The variant function technique has the added advantage that it is readily adapted for use with standard simulation programs, and is ideal for parameter variation studies on existing computer models.

CHAPTER 10

CONCLUSIONS

10.1 Adrenal Modelling

In early attempts to model the adrenal gland, particular emphasis was placed on the overshoot of cortisol secretion following the start of an ACTH infusion. Mechanisms which could reproduce this response were postulated, and potential sites for such mechanisms were searched for, always without success. As knowledge of the biochemistry of the gland improved, models of known mechanisms were investigated to see whether they responded in the same way as the gland. This later approach has been far more successful, although now, three models of the gland are capable of reproducing the observed response, while each model is based on an entirely different mechanism.

The Koritz-Hall model, developed by Urquhart (1970), has been invalidated by evidence that ACTH acts prior to pregnenolone in the biosynthetic pathway (Hall and Young, 1968). The other two models, which are developed in chapter 4.0, require further experimental work to determine whether the biochemical mechanisms on which they are based, occur in the gland to the level suggested by the models. Thus, a closer study of the short term changes of stored cholesterol in adrenal tissue is needed to check the validity of the stored cholesterol model. The model derived from Berger's proposal can be tested by measuring the secretions of DHEA and its metabolites from the adrenal gland.

If DHEA secretion has two phases - a rapid initial rise followed by a slower secondary rise - then the dynamics of the Berger model should be investigated more thoroughly.

To explain the secretion of cortisol at high levels of ACTH stimulation, all three models described in chapter 4 require a saturating mechanism in the later stages of the biosynthetic path by which cortisol is produced. The site of this saturation in the gland has not been found by experiment. It may be independent of the mechanism which produces the overshoot, as it is incorporated into the three models, or alternatively, may be part of the overshoot mechanism itself.

The modelling attempts described in chapter 4 are based more on theory than on hard facts, suggesting that much experimental work is needed to determine the form of the mechanism which controls cortisol secretion during stimulation by ACTH.

Until new evidence or a new theory is available, the modelling approach can offer little to decide the true functioning of the adrenal gland. However, models can point out the weak areas of our knowledge, as is shown in chapter 4.

10.2 ACTH Measurement

The measurement of ACTH using radioimmunoassay is sufficiently sensitive to show that real differences in the concentration of ACTH between the adrenal artery and vein do occur. These results which are described in chapter 5, support the theory that ACTH is bound to, or is otherwise

destroyed within adrenal tissue. However, as the measurement procedure may not measure specifically biologically active ACTH, the true magnitude of the ACTH loss is not determined by these experiments.

The proposal of Bessar et al. (1971), that ACTH radioimmunoassay measures both ACTH and a biologically inactive fragment of ACTH, is supported in chapter 6. A two compartment model of the proposed mechanism is developed and tests made on the model using data from the Lincoln experiments and using published data, show the original proposal to be an analytically plausible explanation of the measurement anomalies. However, the model does not rely on actual fragmentation of the ACTH molecule; any mechanism which renders the ACTH molecule biologically inactive without affecting its immunoreactivity, is covered by the model and hence could explain the measurement differences.

The ACTH fragmentation model can be used to gain estimates of biologically active ACTH concentrations from measurements of immunoreactive ACTH in some experimental situations. However, the analytical procedure used is not readily adapted to automatic analysis because a number of problem situations can arise, particularly when fitting the model to the data.

The model cannot be used to predict active ACTH concentrations when the number of data are small, but it does show the danger in using such data in clinical work. Radioimmunoassay can indicate an ACTH concentration between two and five times the true value, and even on a comparative

basis, the same immunoassay value can result from bioactive ACTH concentrations varying by a factor of two.

The repeatability and relative ease of use of radioimmunoassay techniques make application of this procedure for ACTH measurement more suitable to laboratory implementation than current bioassay techniques. However, before the need to use bioassay procedures can be eliminated, improvement must be made to the technique itself, or to the antibodies used.

10.3 Simulation Languages

Simulation languages are a convenient tool for investigating the solutions to the differential and algebraic equations which comprise a model. Although the numerical procedures used to solve differential equations are computationally inefficient when compared with the direct solution techniques, which can be used on linear models, the loss of efficiency is compensated for by the ease with which simulation languages can be programmed.

Simulation langauges are most useful where relatively few simulation runs are to be performed. For large numbers of simulation runs, with particular sets of parameter values, other methods of analysis are preferable to minimize the computation time. In these cases, special purpose digital computer programs, or hybrid computer programs provide significant speed advantages compared to standard digital simulation techniques. Thus, for the procedures used to estimate the model parameter values in chapter 6, the model

solutions are generated by direct algebraic techniques. To use numerical integration here would increase the computation time by nearly tenfold.

The models which are discussed in chapters 3, 4 and 6 were studied using SIMUL8, a general purpose simulation language described in chapter 7. SIMUL8 embodies all of the main features of simulation languages such as DSL/90 (Syn and Linebarger, 1966) and 360-CSMP (Brennan and Silberberg, 1968), while providing the user with interactive features using a visual display screen and an active teletypewriter. Direct interaction allows the simulator to gain a feel for his model more quickly than where the simulation is performed in the "batch" mode as with DSL/90.

The translation of the model equations into the SIMUL8 language is simple and easy to learn, because of the similarities between the SIMUL8 coding and the model equations.

In the DSL/90 language, the model equations are first passed through a sorting algorithm which adjusts their order to ensure that each variable is assigned a value before it is used. Such a sorting algorithm is an unnecessarily complicated procedure within the concept of SIMUL8, so a new checking method is provided to check the equation order at run time, and to subsequently warn the operator of any errors that have occurred. It is then up to the operator to modify the equation order with the editing programs.

The versatility of SIMUL8 is reflected in the variety of models which have been successfully studied with it, and also by the number of modellers who have used the language.

10.4 Magnetic Tape Controller

The magnetic tape controller, which is described in chapter 8 has provided significant benefits to this research, and to the research of others in the Electrical Engineering Department, University of Canterbury. These benefits have been due to the effective increase in the available disc storage, and to the reduced effect on other users of system failures caused by either hardware faults or accidental misuse of the disc operating system. It has allowed larger programs (such as SIMUL8) to be run on the computer and has made the implementation of such programs quicker.

The concept of the "data present" bit which is introduced in section 8.5, is new, and its application allows particularly simple driving software. Without it many more housekeeping operations would have to be performed by this software.

The flexibility of the controller is shown by its use both as a disc backup storage device and, for a short period while the disc was being serviced, as a primary storage device.

10.5 Parameter Variations

In chapter 9 a preliminary study is made into the methods by which the solutions to sets of linear differential equations with randomly distributed parameters, can be studied statistically to determine the mean solution and the variation of the solutions about the mean.

Of the three techniques described only two appear to be useful computationally. The Monte Carlo technique has advantages where a hybrid computer is used, and the variant function technique where the number of parameters is large. However more study is required to determine how the two methods compare computationally under situations of model size, number of parameters etc.

The variant function technique is more convenient to use than the Monte Carlo method as it is readily incorporated into existing simulation studies and into standard simulation languages. However, further study is necessary in the following directions to improve the variant function technique and to allow it to be used in more general situations.

I The development of the technique is concentrated on linear systems. Because nonlinear systems are so common in biological models, the implications of applying the technique to nonlinear systems should be investigated.

II The selection of the variant function periods (T_{ij}) has been fairly arbitrary here, and further study is required to determine more rigorous ways to select these. It may be possible to use the same period for two variant functions which describe areas of the model remote from each other.

III It may not be necessary to transform model parameters which are relatively small in size, because their effect on model solutions is transitory by nature and may have little effect on the solutions of

interest. Parameters of this type should be studied to see whether anything is gained by transforming them into variant functions.

IV Finally, a study should be made into general methods of incorporating the variant function technique into existing computer simulations and extracting the mean solutions and variations of the solutions about the mean. In this study particular attention should be paid to points I, II and III above.

APPENDIX 1

Function Minimization Using the Pattern Search Algorithm

The "pattern search" function minimization procedure devised by Hooke and Jeeves (1961) is reported to be well suited to non-linear curve fitting problems involving the minimization of a sum of squares (Wilde, 1964). A complete account of the algorithm is given by both Hooke and Jeeves (1961), and Wilde (1964). A brief outline of the algorithm and flow charts of the procedures (as they have been implemented on the EAI 640 computer by Fromm, 1970) used in this thesis are now presented.

Pattern search is a direct search procedure (Hooke and Jeeves notation, 1961) for finding the position of the minimum of a function of a number of variables. In contrast to the "method of steepest descent" (cf. Elgerd, 1967), direct search procedures do not require the derivatives of the function to be calculated. Instead, in direct search procedures, new moves are predicted solely from the success of previous moves in the parameter space, in reducing the function value.

In general direct search procedures rely on the premise that "if a move in a particular direction of the parameter space is successful, then a further move in a similar direction is also likely to be successful". Similarly, if a move is a failure, further moves in that direction are unlikely to succeed.

Pattern Search - A Direct Search Strategy

In the pattern search algorithm, two types of moves in the parameter space are used. The first type, the exploratory move, is used to investigate the region about a point in the parameter space. The exploratory move determines a direction which reduces the function value. This direction information is used to make a "pattern" move, the second type of move. The direction of this pattern move is then updated by a further exploratory move, and a further pattern move implemented. The pattern move requires only one function evaluation per move, and therefore makes the fastest progress in the search for the minimum. Exploratory moves require between n and 2n evaluations, where n is the number of independent variables in the function to be minimised. The exploratory move and pattern move algorithms are now described in more detail.

The Exploratory Move

In an exploratory move each of the independent variables $(x_1, x_2, \ldots, x_n)$ of the function $F(x_1, x_2, \ldots, x_n)$ are considered in turn. The first variable x_1 is increased by a small amount Δ_1 and the function value $F(x_1+\Delta_1, x_2, \ldots, x_n)$ compared with that of the initial base point $(x_1, x_2, \ldots, x_n)$. If the move is successful, i.e. if the function has been reduced by the move, the new coordinate value is retained and the next variable is processed. If the move is a failure, the variable value is reduced to $x_1 - \Delta_1$ and the function is again evaluated. If this

second move is successful the value $x_1 - \Delta_1$ is retained and the next variable processed. If the second move also fails the variable is restored to its original value (x_1), and the next variable is processed. This procedure is repeated for each of the n variables.

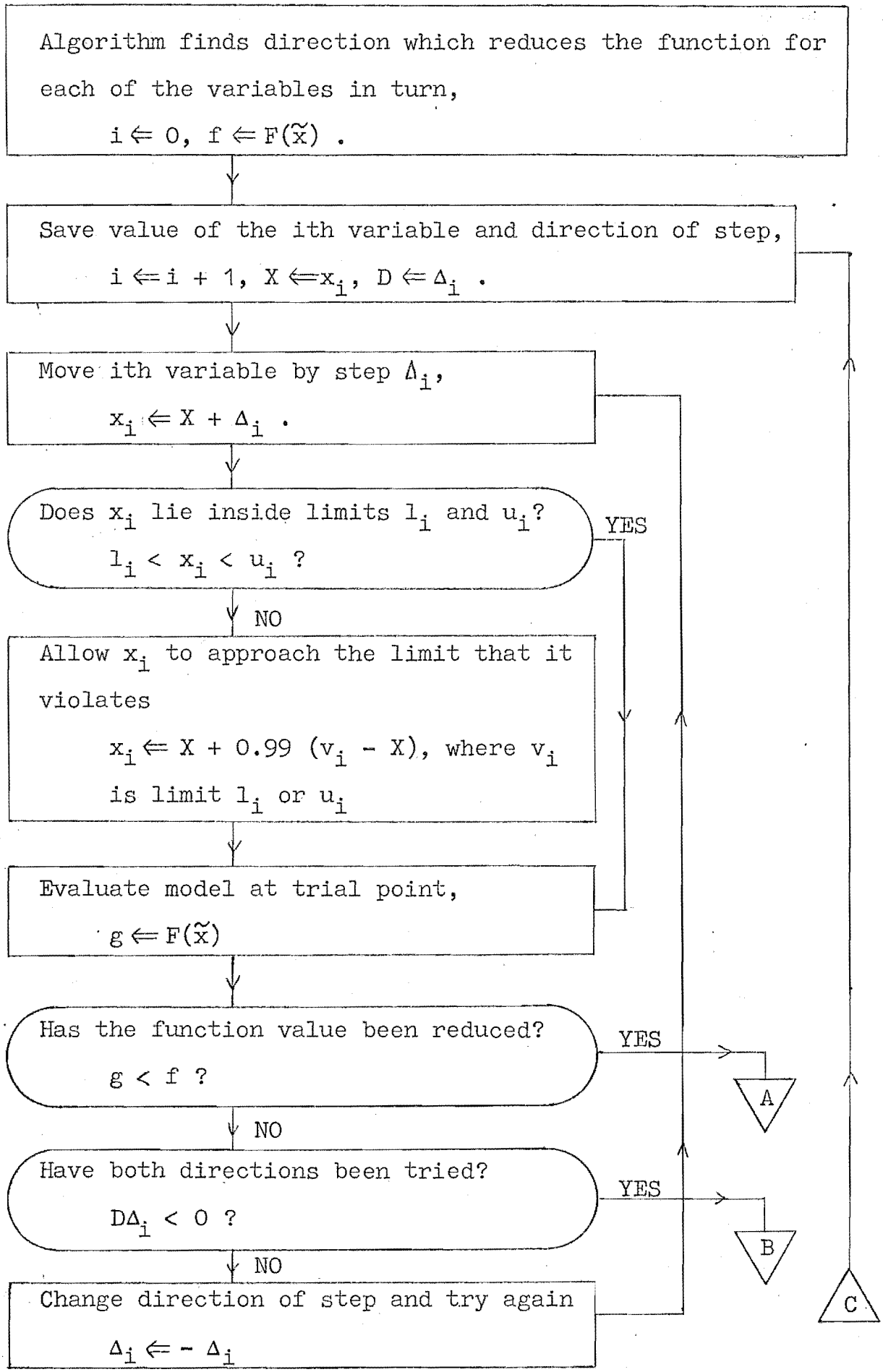
If one of the two moves for a given variable x_i is successful the steplength Δ_i for that variable is increased by a factor α so that on subsequent exploratory moves larger steps are taken. Conversely if both moves fail the steplength is reduced by a factor β so that on subsequent exploratory moves a finer search is made. So that the number of function evaluations is minimized, the direction of the move made by each variable is saved, so that this direction will be tried first on subsequent exploratory moves.

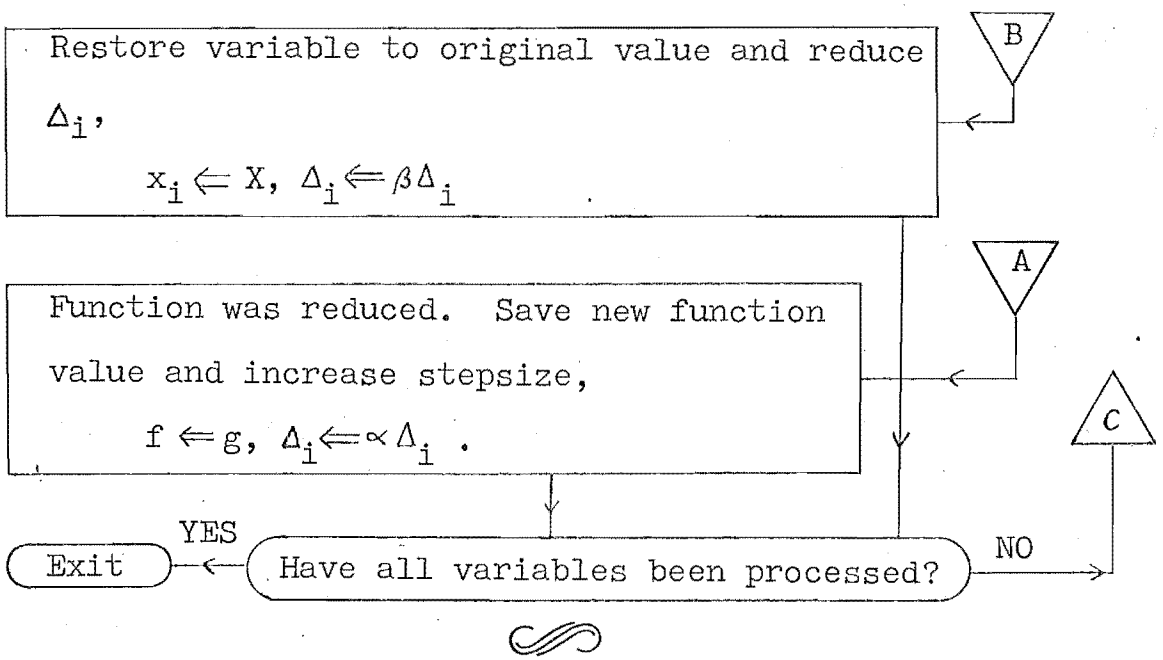
Upper and lower values for each variable are supplied to the algorithm by the user to allow the exploration to be confined to a particular region of the parameter space. Moves made by the algorithm which violate these limits are reduced in size in such a way that the limit points are approached gradually.

A detailed flow chart of the exploratory move algorithm is shown in figure A.1.

Examples of exploratory moves, in two variable function space, are shown in figure A.2.

Figure A.1 The exploratory move algorithm.





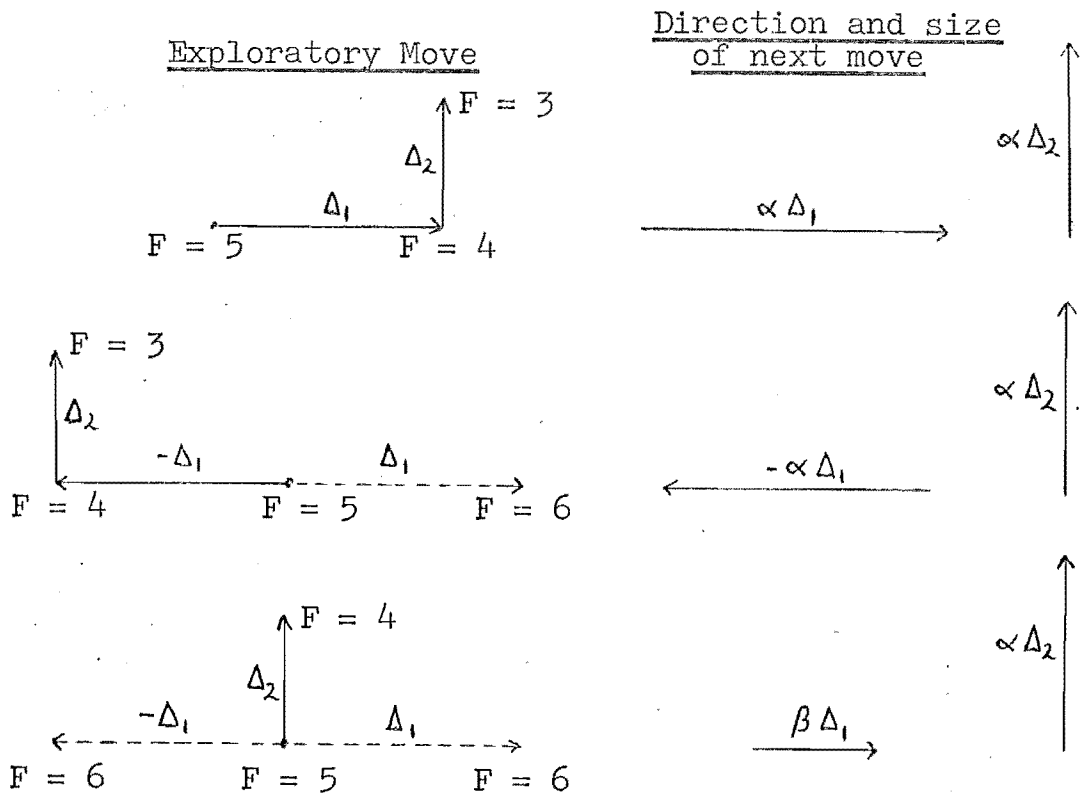


Figure A.2 Examples of exploratory moves in function space $F(x_1, x_2)$ showing end positions of moves, and effects of a success and a failure on subsequent step size.

The Pattern Move

When an initial exploratory move about the point $\tilde{x}_0$ finds a point $\tilde{x}_1$ which reduces the function value, a pattern move of length $2(\tilde{x}_1 - \tilde{x}_0)$ is made from the base boint $\tilde{x}_0$. The pattern move thus reaches a point $\tilde{x}_2$, where

$$\tilde{x}_2 = 2\tilde{x}_1 - \tilde{x}_0 \qquad (A.1)$$

If the function is reduced by this pattern move a further pattern move is initiated. But first, the region about $\tilde{x}_2$ is explored using the exploratory move algorithm, to allow an improved direction for the pattern move to be found. The second pattern move is made to the point $\tilde{x}_4$, where

$$\tilde{x}_4 = 2\tilde{x}_3 - \tilde{x}_1 \ , \qquad (A.2)$$

and $\tilde{x}_3$ is the position determined by the second exploratory move. This process is repeated until a pattern move fails to reduce the function value. At this point a new pattern must be initiated by making an exploratory move from the previous point with the lowest function value.

The speed with which pattern search approaches the minimum is a direct function of the size of the steps taken in the variable space. These step sizes are controlled by the success or failure of the exploratory moves. Thus, as more successful moves are made within the exploratory move procedure, the step sizes increase and so does the size of the pattern moves.

Because pattern search essentially moves along the valleys of the function, which are basically one dimensional (cf. Wilde, 1964), the computation time increases as the first power of the number of variables. In contrast the computation time of classical function minimization techniques increases as the cube of the number of variables (Wilde, 1964).

An example of a pattern search is shown in figure A.3. The detailed flow chart for the technique is shown in figure A.4.

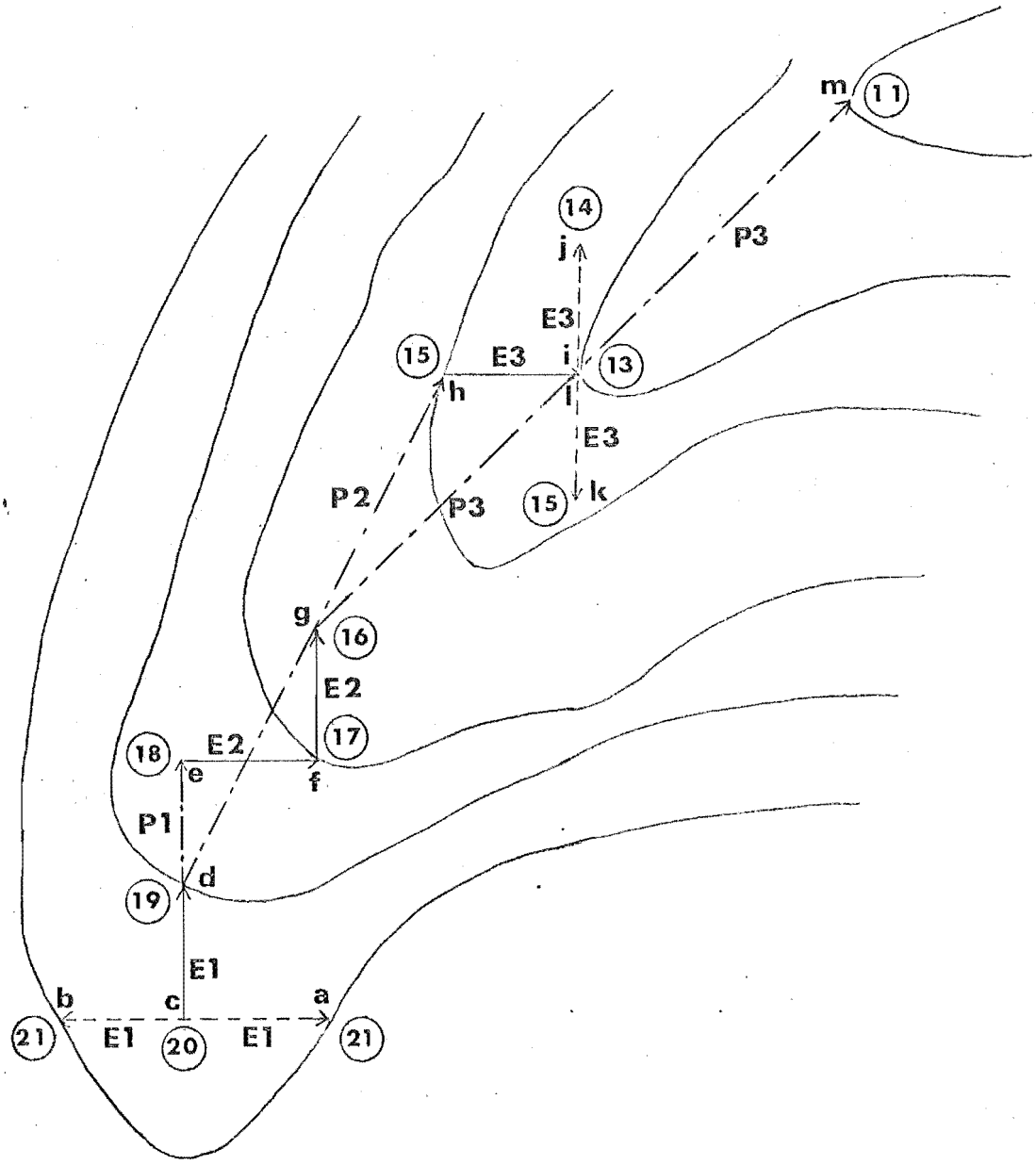
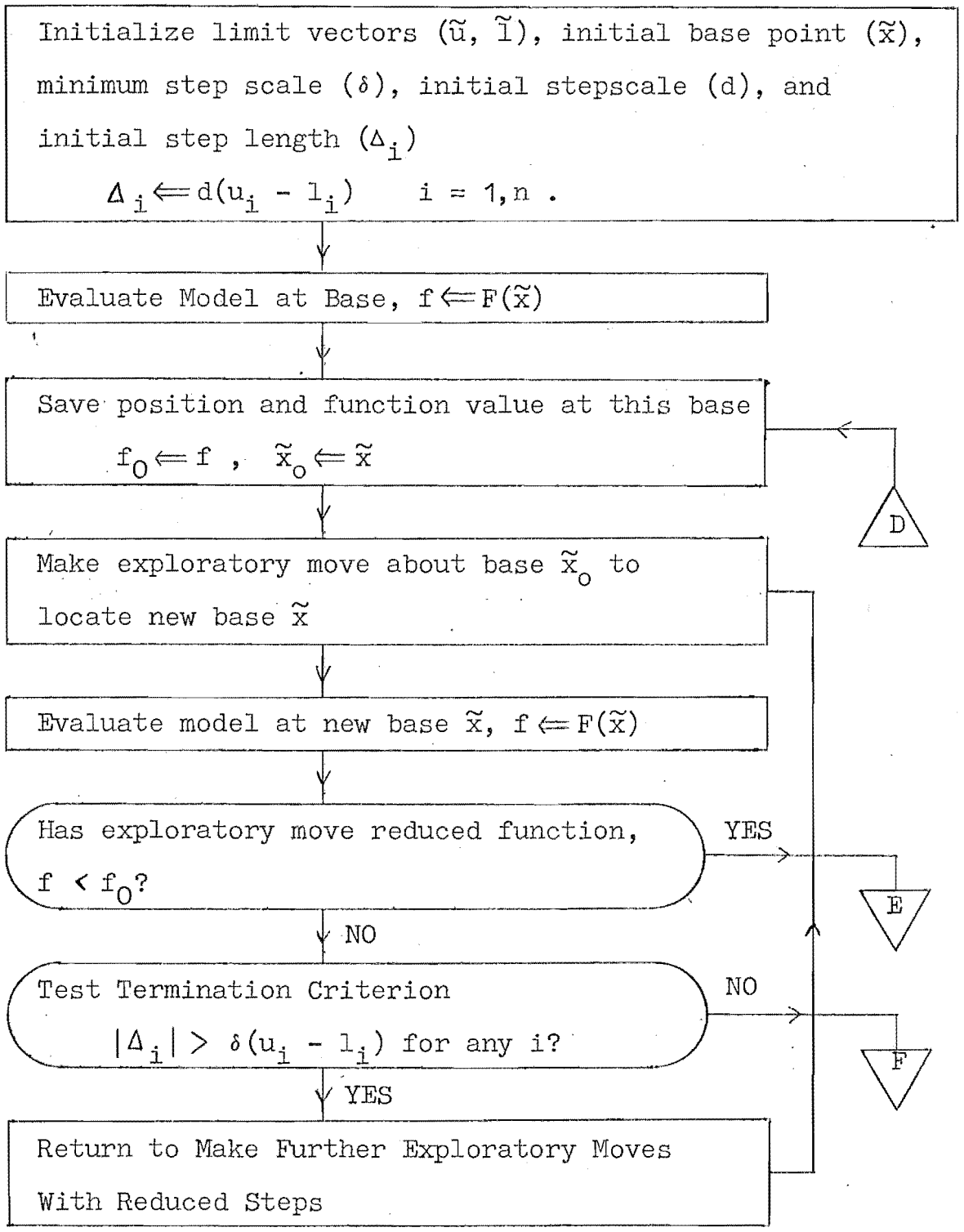
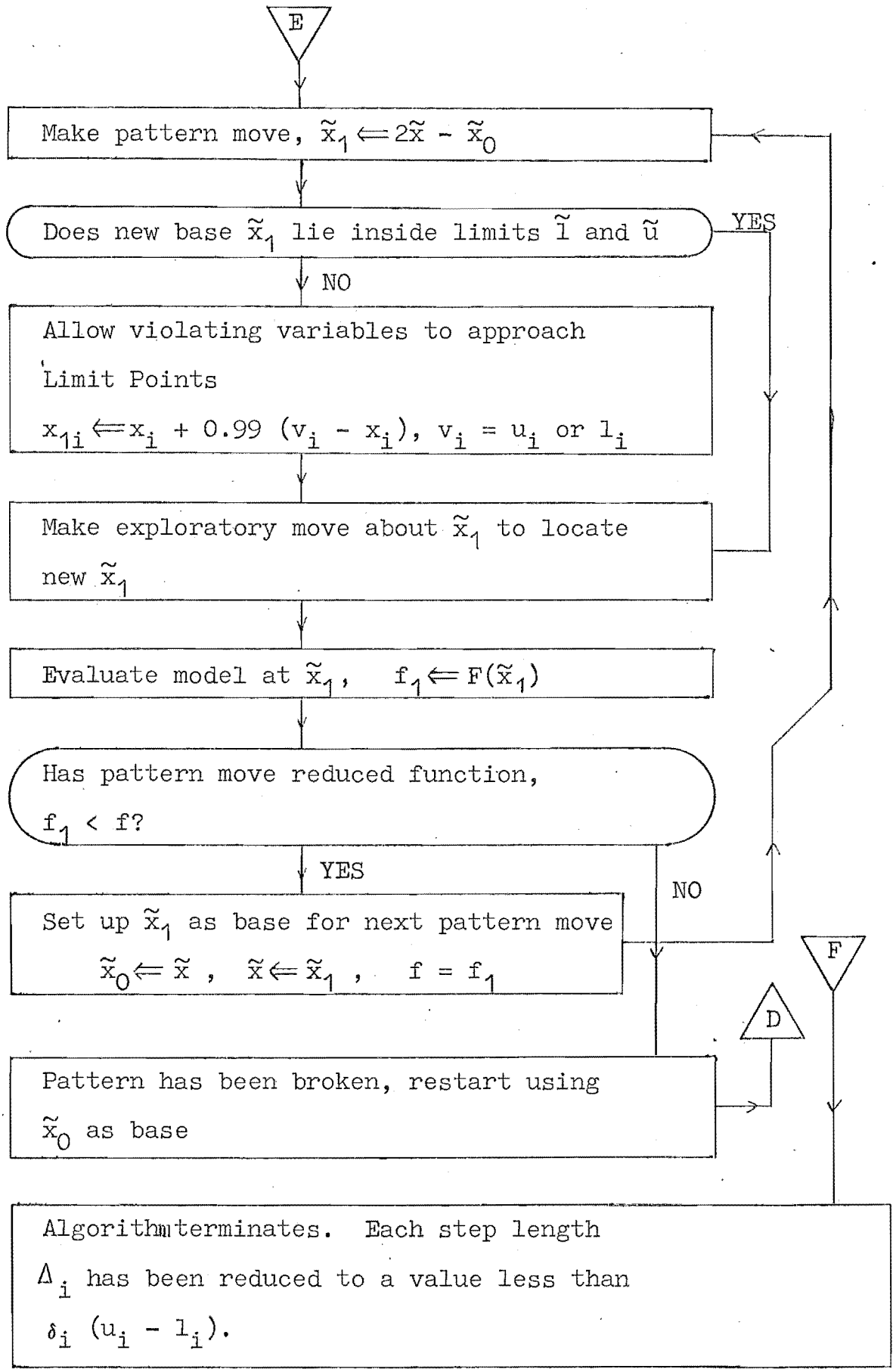


Figure A.3 Example of the pattern search technique, showing how moves adjust to orientate along the valley floor. The effect of the stepsize scaling vectors α and β is not shown in the figure. Circled numbers are function values at the associated points. E_n is the nth exploratory move. P_n is the nth pattern move. ——→ Exploratory move which reduces function. - - - - - → Exploratory move which fails. ——- - —— → Pattern move. Alphabetic sequence a - m shows sequence of moves.

Figure A.4 Flow chart of the pattern search algorithm.




E
Make pattern move, $\tilde{x}_1 \Leftarrow 2\tilde{x} - \tilde{x}_0$
Does new base $\tilde{x}_1$ lie inside limits $\tilde{l}$ and $\tilde{u}$
YES
NO
Allow violating variables to approach Limit Points
$x_{1i} \Leftarrow x_i + 0.99\ (v_i - x_i)$, $v_i = u_i$ or l_i
Make exploratory move about $\tilde{x}_1$ to locate new $\tilde{x}_1$
Evaluate model at $\tilde{x}_1$, $f_1 \Leftarrow F(\tilde{x}_1)$
Has pattern move reduced function, $f_1 < f$?
YES
NO
Set up $\tilde{x}_1$ as base for next pattern move
$\tilde{x}_0 \Leftarrow \tilde{x}$, $\tilde{x} \Leftarrow \tilde{x}_1$, $f = f_1$
F
D
Pattern has been broken, restart using $\tilde{x}_0$ as base
Algorithm terminates. Each step length Δ_i has been reduced to a value less than $\delta_i\ (u_i - l_i)$.

Termination Criterion

As the position of the function minimum is approached the sizes of the moves made are reduced by the exploratory move procedure, allowing the minimum to be more accurately found. This stepsize reduction occurs when the optimum point is stepped over by the algorithm and, unlike the method of steepest descent, is not directly related to the slope near the optimum point.

In this way the speed of movement is not reduced in valleys with a small gradient.

The algorithm terminates when the stepsize has been reduced to a predetermined size. Thus

$$\Delta_i < \delta(u_i - l_i) \text{ , for all i.} \qquad \text{(A.3)}$$

This convergence test is made if the first exploratory move in a new pattern fails to reduce the function.

APPENDIX 2

Data Interpolation Using Akima's Procedure

The interpolation procedure devised by Akima (1970) is based on piecewise continuous polynomials of order three, at most, having continuity in the interpolating function and its derivative over the extent of the data. The interpolating function is

$$y = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad (A.4)$$

where

$$x_i \leq x \leq x_{i+1}, \quad (A.5)$$

and

$$(x_i, y_i),\ i = 1, 2, 3, \ldots, n \quad (A.6)$$

are the coordinates of the function to be interpolated.

Akima derives an expression for an estimate of the gradient of an interpolating function at a point (x_i, y_i), given the coordinates of two points on each side of the point (x_i, y_i)(viz. (x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}) and (x_{i+2}, y_{i+2})). This expression is

$$y_i' = \frac{m_{i-1} \cdot |m_{i+1} - m_i| + m_i \cdot |m_{i-1} - m_{i-2}|}{|m_{i+1} - m_i| + |m_{i-1} - m_{i-2}|}; \quad 2 < i < n - 1 \quad (A.7)$$

where y_i' is the gradient of the interpolating function at the point (x_i, y_i), and

$$m_j = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} . \qquad (A.8)$$

Using equation (A.7) the gradients at all data points excluding two points at each extremum can be found. Estimates of the gradients at the extreme points may be made by fitting a quadratic expression of the form

$$y = p + qx + rx^2 \qquad (A.9)$$

to the first three points and to the last three points. The gradient of the interpolating function at the first two points may be found by solving equation (A.9) for q and r using the data (x_1, y_1), (x_2, y_2), (x_3, y_3), and then evaluating the derivative of equation (A.9),

$$y' = g + 2rx \qquad (A.10)$$

at (x_1, y_1) and (x_2, y_2).
This procedure yields

$$y_1' = m_1 - m_2 + \frac{y_3 - y_1}{x_3 - x_1} \qquad (A.11)$$

$$y_2' = m_1 + m_2 - \frac{y_3 - y_1}{x_3 - x_1} . \qquad (A.12)$$

Similarly, by using the same procedure on the last three data points (viz. (x_{n-2}, y_{n-2}), (x_{n-1}, y_{n-1}), and (x_n, y_n)) the gradients y'_{n-1} and y_n' can be shown to be

$$y'_{n-1} = m_{n-2} + m_{n-1} - \frac{y_n - y_{n-2}}{x_n - x_{n-2}} , \qquad (A.13)$$

and

$$y'_n = m_{n-1} - m_{n-2} + \frac{y_n - y_{n-2}}{x_n - x_{n-2}} . \qquad (A.14)$$

The unknowns a_i, b_i, c_i and d_i in equation (A.4) can now be found using the original data values (x_i, y_i) and (x_{i+1}, y_{i+1}), and the gradient estimates from equations (A.7), (A.11), (A.12), (A.13) and (A.14), for each interval $1 \leq i \leq n - 1$ by substituting the data values into equation (A.4), the gradient estimates into the derivative with respect to x of equation (A.4), and solving.

This yields

$$a_i = y_i \quad , \tag{A.15}$$

$$b_i = y_i' \quad , \tag{A.16}$$

$$c_i = (3m_i - 2y_i' - y'_{i+1})/(x_{i+1} - x_i) \quad , \tag{A.17}$$

and

$$d_i = (y_i' + y'_{i+1} - 2m_i)/(x_{i+1} - x_i)^2 \quad . \tag{A.18}$$

Thus a value for an interpolated point (x, y) may be obtained using equation (A.4) with the constants a_i, b_i, c_i and d_i defined as in equations (A.15), (A.16), (A.17) and (A.18).

Akima (1970) shows that the interpolation method presented here gives a more natural looking interpolating function than many of the commonly used interpolation techniques. Akima cites the following techniques

(a) Polynomial of order n - 1,

(b) Fourier cosine,

(c) Cubic spline function,

(d) Osculatory,

(e) Manually drawn curves,

and shows by example that his method produces a curve very similar in form to the manually drawn curves.

Computationally, Akima's technique is considerably faster than the cubic spline but not quite as fast as the osculatory technique.

APPENDIX 3

Solution of a Set of Linear First Order Ordinary Differential Equations Using the Matrix Exponential

Given the linear, first order, ordinary differential equations of the form

$$\frac{d}{dt} y(t) + Ay(t) = f(t) \quad , \tag{A.19}$$

we show the solution for y(t) to be

$$y(t) = e^{-At} y(0) + \int_0^t e^{-A(t-\tau)} f(\tau)\, d\tau \quad , \tag{A.20}$$

where e^{-At} is the state transition matrix defined as

$$e^{-At} = \sum_{i=0}^{\infty} \frac{(-At)^i}{i!} \quad ; \tag{A.21}$$

$$(-At)^0 = I \quad , \tag{A.22}$$

the identity matrix (cf. McCausland, 1959).

The matrix exponential e^{-At} has the properties

$$e^{At} \cdot e^{-At} = I \quad , \tag{A.23}$$

and

$$\frac{d}{dt} e^{-At} = -Ae^{-At} \quad , \tag{A.24}$$

(cf. McCausland, 1969).

The solution to equation (A.19) has the form

$$y(t) = e^{-At} x(t) \quad , \tag{A.25}$$

where x(t) is a function yet to be defined. We start by

finding x(t) by substituting equation (A.25) into equation (A.19), noting that

$$\frac{d}{dt} y(t) = \dot{y}(t) = e^{-At} \dot{x}(t) - Ae^{-At} x(t) . \qquad (A.26)$$

Thus,

$$f(t) = e^{-At} \dot{x}(t) , \qquad (A.27)$$

and

$$\dot{x}(t) = e^{At} f(t) . \qquad (A.28)$$

Integrating equation (A.28) yields

$$x(t) - x(0) = \int_0^t e^{A\tau} f(\tau) \, d\tau , \qquad (A.29)$$

where τ is a dummy variable of integration. By substituting equation (A.29) into equation (A.25) we get

$$y(t) = e^{-At}\left[\int_0^t e^{A\tau} f(\tau) \, d\tau + x(0)\right] . \qquad (A.30)$$

Also, from equation (A.25)

$$x(0) = e^{A0} y(0) = y(0) \qquad (A.31)$$

The solution to equation (A.19) is therefore

$$y(t) = e^{-At} y(0) + \int_0^t e^{-A(t-\tau)} f(\tau) \, d\tau . \qquad (A.32)$$

We also note that if

$$f(t) = 0 , \qquad (A.33)$$

then

$$y(t) = e^{-At} y(0) . \qquad (A.34)$$

By using a similar technique to that used to solve equation (A.19), we can show that the solution to the equation

$$\frac{d}{dt}\,\eta(t) + \frac{d}{dt}\,\mathfrak{A}(t)\,\eta(t) = f(t) \quad , \tag{A.35}$$

is

$$\eta(t) = e^{-\mathfrak{A}(t)}\left[\int_0^t e^{\mathfrak{A}(\tau)}\, f(\tau)\, d\tau + \eta(0)\right] . \tag{A.36}$$

As before, let

$$\eta(t) = e^{-\mathfrak{A}(t)} \cdot x(t) . \tag{A.37}$$

Then

$$\frac{d}{dt}\,\eta(t) = e^{-\mathfrak{A}(t)} \cdot \dot{x}(t) - \dot{\mathfrak{A}}(t)\, e^{-\mathfrak{A}(t)}\, x(t) . \tag{A.38}$$

Substituting equations (A.37)and (A.38) into equation (A.35) and rearranging yields

$$x(t) - x(0) = \int_0^t e^{\mathfrak{A}(\tau)}\, f(\tau)\, d\tau . \tag{A.39}$$

Using equation (A.37) and (A.39), the solution $\eta(t)$ is found to be

$$\eta(t) = e^{-\mathfrak{A}(t)}\left[\int_0^t e^{\mathfrak{A}(\tau)}\, f(\tau)\, d\tau + \eta(0)\right] . \tag{A.40}$$

APPENDIX 4

SIMUL8

A SIMULATION SYSTEM FOR THE EAI 640

(Users Manual)

A.E. McKinnon and R.B. Jordan

March 1973

CONTENTS

Contents (cont.)

List of Appendices

1.0 INTRODUCTION

This set of programs, SIMUL8, has been designed to enable the user to investigate the properties of simple continuous models.

The model may be expressed in the form of a set of ordinary first order differential equations, or simply as an algebraic expression which the user wishes to evaluate for various values of an independent variable. In both cases the interactive aspects of such an investigation are provided for, using the display screen and active teletype in the EAI 640 computing system.

1.1 Pertinent Documents

The following two manuals should be used in conjunction with the SIMUL8 system manual.

1. EAI 640 Fortran Manual and Notes.

2. EAI 640 Disc Operating System Manual.

2.0 PROGRAM STRUCTURE

The program consists of a number of utilities made available to the user via a command package accepting control from the teletype. Basic to the system is a utility which solves numerically a number of differential equations by incrementing time in small steps and computing new values for the equation solutions from their derivatives which are supplied by a user written subroutine. Time is regarded as the model independent variable. If there are no differential equations in the model, time is incremented in the same way, but the integration algorithm is by-passed. The user's equations are supplied to the package in the form of a FORTRAN subroutine previously compiled. This program is loaded by executing the unsatisfied Program Loader which forms part of SIMUL8.

2.1 Data Structure

All data associated with the user's model is contained in four separate COMMON areas as follows:

COMMON EQTNS	storage of model dependent variable values.
COMMON DERIV	storage of the derivatives corresponding to EQTNS
COMMON YSTART	storage of initial conditions corresponding to EQTNS.
COMMON MODPAR	storage of model parameters.

2.2 Program Control

The typing of the message "X←" indicates that SIMUL8 is ready to accept a command from the teletype. The commands, which consist of 2 characters without a carriage return are fully summarised in Appendix V.

3.0 SPECIFYING THE MODEL

To set up a system to be studied, the user must write a FORTRAN subroutine named DIFEQN in which the model equations are specified as standard FORTRAN statements. For any differential equations an explicit expression for the first derivative of each dependent variable must be given.

It is important to ensure that all quantities on the right hand side of an expression have been assigned values prior to execution of that expression.

For example, the statements,

```
YDOT = R
   R = - A * Y
```

where YDOT represents the derivative of Y with respect to time, are in the incorrect order and will cause errors. (Section 3.4).

In the demonstration problem of Appendix III, an example is given of a DIFEQN program for a second order RLC electrical circuit. This should be studied in conjunction with the following rules.

3.1 Detailed Rules

For the correct operation of the simulation, it is important that the following rules be adhered to:-

(a) The model dependent variables (i.e. differential equation solutions) must be stored in the EQTNS common block.

(b) The number of model differential equations must correspond to that given during the initialisation procedure (see section 11.0) or when using the NE command (see section 4.1.1).

(c) The derivatives must be stored in the DERIV common block in the same order as the corresponding quantities in EQTNS common.

(d) Model constants over which the user wants to have external control must be stored in the MODPAR common block.

3.2 Function Blocks and Subroutines

The user may incorporate in his DIFEQN program calls or references to any subprograms which he has written to perform auxilliary calculation. Such subprograms would then be regarded as part of the user's model.

A number of standard function blocks are provided in SIMUL8 to enable easy incorporation of such things as pulse and deadspace effects. Details of these are given in Appendix II.

3.3 Output Considerations

As will be explained in section 5.1.1, only quantities in the EQTNS common block can be sampled during the course of a simulation for subsequent output. Therefore, if the user wishes to sample a variable which is not a differential equation solution, such a variable must also be stored in the EQTNS common area.

If the user's model has n differential equations, the first n cells of EQTNS must be allocated for differential equation solutions. Any of the other cells in this common block may be used for storing auxilliary variables the user may want to output.

3.4 Equation Order and Undefined Constants

If any quantity used in calculation during a simulation has not been previously defined the message,

UNDEFINED CELL AT XXXXX

will be typed and computation will continue .XXXXX is the octal location containing the undefined floating point number.

The following lists the possible conditions under which this error could arise. The user's DIFEQN subroutine is loaded at '21537 and the addresses of the common blocks are given in Appendix 1.

1. Initial conditions not defined (this could mean that NE is set too large) -XXXXX will be in the EQTNS common block.

2. Model parameters not defined -XXXXX will be in the MODPAR common block or in the users DIFEQN subroutine.

3. Equations out of order -XXXXX will be in the EQTNS common block or in the users DIFEQN subroutine.

4. Differential equations omitted (this could mean NE is set too large) -XXXXX will be in the DERIV common block.

In cases 1 and 3 the error message will be typed only once when simulation is first attempted: subsequent results may be inaccurate.

Depending on the nature of the error it may be necessary to make the correction in the user's source version of the DIFEQN subroutine.

4.0 MANIPULATION OF PARAMETERS

Apart from the initial setting up procedure when the program asks for the parameter settings vital to the function of the simulation, parameter values will not be changed unless the user types the appropriate commands. Parameters may be set from the teletype and provision is made for the setting of model parameters and initial conditions from papertape.

4.1 Teletype

When interrogating parameter values from the teletype, the user types a command which specifies the particular parameter, whose value is then displayed. A new value may then be input by the user.

4.1.1 Interrogation Commands

There are certain quantities which are investigated by a specific command, the response being an immediate display of the

present value for that quantity and a request for a new value. Typing a carriage return causes the current value to be retained. The commands for this category of parameter are:

Command	Parameter
NS	Max. number of integration steps.
NE	Number of differential equations.
DT	Integration step length.
TF	Simulation finish time.
TR	Trace time.

The following commands are provided for the interrogation of groups of quantities such as the model parameters in the MODPAR common block.

Command	Parameter
MP	MODPAR common area
YØ	YSTART common area
YY	EQTNS common area

It is not possible to interrogate the DERIV common block.

Following one of these last three commands the program will wait for input in one of the following formats:-

(a) N:

or (b) N,M:

The response is as follows:-

(a) The Nth element of the array corresponding to the command, is typed out and the program waits for a new value to be entered. If the user wishes to retain the current value a carriage return should be typed. The program then waits for a new parameter within that array to be specified for interrogation.

(b) Parameters N to M inclusive of the array corresponding to the command are dumped on the teletype. The program then waits for a new parameter within that same array to be specified for interrogation.

4.1.2 Numerical Format

For all numerical input from teletype or papertape, the field length is variable being determined by the terminating character CR. Quantities specified under the NS or NE commands (section 4.1.1) must be typed as integers, whereas all other commands specify real numbers which may be input in FORTRAN E or F format or as integers. For example .00105E5, 105.0 and 105 are equivalent real number inputs.

4.1.3 Rubout

Typing the "RUBOUT" character causes the previous character to be ignored except when "RUBOUT" is the first character of a line. In this case the RUBOUT will be ignored thus preventing deletion of characters beyond the beginning of the line.

4.1.4 Errors

Any errors detected during numerical input will cause a '?' to be typed and the number may be entered again.

4.1.5 X Character

The typing of the X character at any stage during teletype input causes control to be immediately transferred to the command input program. A number being typed in will not be processed or stored if the X command is typed before the CR.

Typing X is the means by which control is returned from the parameter interrogation phase to the command input phase.

4.2 Papertape

Model parameters and initial conditions may be set from papertape which has been previously prepared. Provision is also made for the dumping of these parameters onto papertape.

4.2.1 Commands

The four commands associated with this feature are listed below.

Command	Function
RY	Read initial conditions.
RP	Read model parameters.
OY	Punch initial conditions.
OP	Punch model parameters.

The commands RY and RP will read in a tape formatted as described in section 4.2.2 and store the numbers on it sequentially in the YSTART and MODPAR common blocks respectively. A 'FORM' character on the end of the tape is used to terminate reading.

The commands OY and OP allow punching of initial conditions and model parameters. For OY, "NE" initial conditions are punched corresponding to the number of user differential equations, and for OP the user is asked how many parameters are to be punched. For both these commands comments (see section 4.2.2) are punched at the beginning of the tape identifying the simulation and the run number after which punching takes place. Tapes generated with OY and OP commands are suitable for reading using RY and RP commands respectively.

4.2.2 Tape format

The numerical values on the tape should be given as specified in sections 4.1.2 and 4.1.3. They will be stored in the order in which they appear on tape. Any linefeeds generated when the tape is prepared will be ignored.

A 'FORM' character must be placed at the end of a tape to tell the program when the reading is to finish.

A $ character at the beginning of a line will cause the rest of that line to be treated purely as comments and it will be displayed on the teletype.

4.2.3 Errors

If any number field on the tape contains invalid characters, the number will not be processed and nothing will be stored in that cell. A message will appear on the teletype indicating the number of the parameter containing the error and reading will continue.

5.0 GRAPHICAL OUTPUT

To avoid graph scaling problems there is no facility for graphical output during the course of a simulation. Instead the user may cause samples of up to four of the quantities in the EQTNS common block to be stored, during a simulation, for subsequent examination. These samples will be held in core until they are overlaid by the samples from the next simulation. Facility is provided for permanent saving of the samples of one of these variables and user supplied data from papertape. This enables different simulations and other data to be compared. A flow chart describing these processes is shown in Fig. 5.1.

5.1 Preparation of Variables for Plotting

5.1.1 Prepare Command

The prepare command enables the user to inform the program which of the variables in the EQTNS common are to be sampled and stored for later inspection. The command mnemonic is PR. Following this the user should enter up to four numbers corresponding to the variables he wishes to prepare. e.g. PR ← 1, 3, 5, 4 means variables 1, 3, 5 and 4 in EQTNS common are to be sampled during each simulation.

5.1.2 Save Command

By using the "save" command the user can have the samples of any one of the prepared variables transferred to a permanent storage area where they will not be destroyed by a simulation run. This enables the user to keep the results of one simulation for comparison with those from later runs.

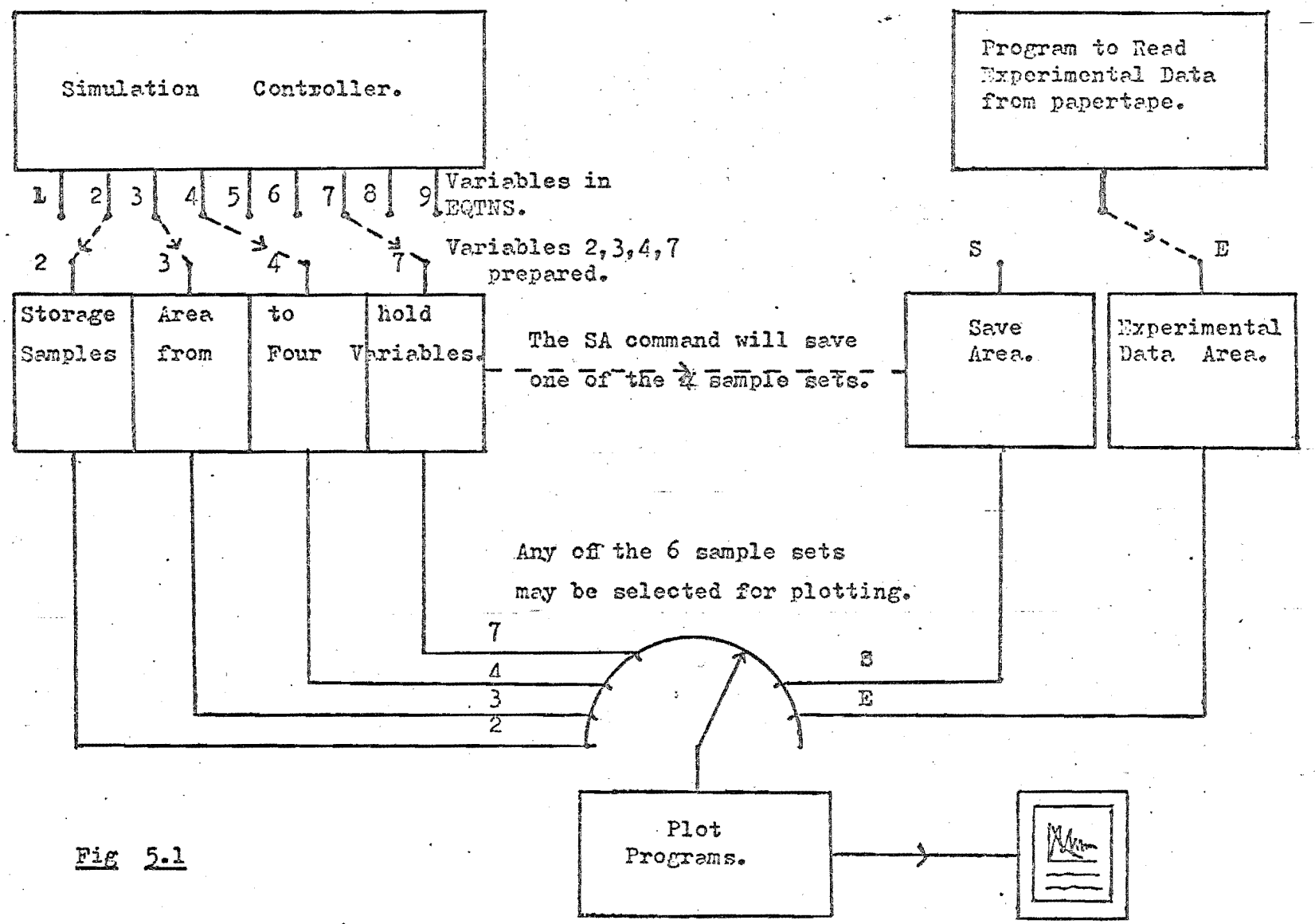

Simulation Controller.
Program to Read Experimental Data from papertape.
1 2 3 4 5 6 7 8 9 Variables in EQTNS.
2 3 4 7 Variables 2,3,4,7 prepared.
S
E
Storage Area to hold Samples from Four Variables.
The SA command will save one of the 6 sample sets.
Save Area.
Experimental Data Area.
Any of the 6 sample sets may be selected for plotting.
7
4
3
2
S
E
Plot Programs.

Fig 5.1

The command format is SA ← N where N is the number of the prepared variable to be saved. These saved samples are subsequently referenced by the letter S rather than the number of the variable from which they were formed. Executing the save command will destroy the previous contents of this permanent storage area.

5.1.3 Experimental Data

By using the ED command (see section 8.0) the user can load experimental data from paper tape into the permanent storage areas of core. The data may be loaded into either the save area or the experimental data area which are referred to by the letter S and E respectively.

5.2 Plotting

There are four commands available to allow the user to plot stored samples on the display screen. All commands have the format DX ← I, J, K.... . DX is the command mnemonic which may be DA, DF, DO, DN. The letters I, J, K etc can be the number of the variable to be plotted, the letter E if the experimental data is to be plotted or the letter S if the saved samples are to be plotted.

The four plotting commands are as follows:-

DA — Erase the screen, draw a new grid and plot, with independent scaling and origin suppression, each of the variables listed.

DF (Family) — Erase the screen, draw a new grid and plot the variables listed using scale factors and origin suppression information calculated from the first variable of the list.

DO (Old Scale) — Plot on the current axes the variables indicated using scaling and origin suppression information from the preceeding curve.

DN (New Scale) — Plot on the current axes, with automatic scaling and origin suppression, the variables listed.

5.3 Plot Errors

A number of error conditions can occur during plotting. If the string of characters following the command cannot be deciphered by the plot programs, a ? will be typed and the user permitted to re-enter the line.

The message "NOT PREPARED" can occur as a result of the following error conditions:

1. No data has been loaded into the experimental data area using the ED command.

2. No data has been loaded into the save area using the SA or ED commands.

3. The variable asked for was not prepared using the PR command before the last simulation run.

5.4 Reading the Plots

To obtain the correct value of the independent variable, the value on the horizontal axis should be multiplied by the scale factor.

For every variable plotted an entry is made in the table of particulars below the graph. Here the plotting symbol is given with:

(a) the number of the variable in the EQTNS common block,

(b) the simulation run number during which it was generated,

(c) ordinate scale factor for that variable,

(d) the ordinate value which corresponds to zero for that variable (i.e. origin).

(e) Maximum value of variable,

(f) minimum value of variable.

To find the actual ordinate of a particular variable, the origin used should be subtracted from the reading on the 0-100 scale, and the result multiplied by the scale factor.

5.5 Origin Suppression

If it is desired to inhibit the automatic suppression of the ordinate origin on the graphs, sense switch C should be pressed.

N.B. With sense switch C pressed, a graph with any negative values will go below the horizontal axis.

6.0 Numerical Output

To obtain numerical output of the data samples stored in core the command LD should be used. The values may be output on the teletype, paper tape punch or display screen.

The user is first asked for the output device number which should be 1, 3 or 5 for Teletype, Display or high speed punch respectively.

The number of the variable to be output is next requested. This may be the number of a prepared variable, the letter S for the save array or the letter E for the experimental data array.

Finally, the user should enter the number of points to be output. The points output will be at equal spacing in time over the simulation time TF. To achieve this it is possible that a few more points than the actual number requested will be output.

In the case of the experimental data array the output points will not necessarily be equally spaced in time, as this array allows points to be stored at non-uniform time intervals.

If the user wants all points to be output he should type 200 for the number of points in the case of the S array or prepared variables, and 50 in the case of the experimental data array.

Dumping onto papertape has an output format suitable for reading in using the ED command (see section 8.0).

If an abnormal termination of a dump is required, sense switch A should be depressed.

7.0 SIMULATION CONTROLS

7.1 Step Length

The routine for solving the differential equations uses a fixed step length algorithm. The user may set this step interval using the DT command. A step length of about one tenth of the smallest model time constant is usually satisfactory, however successive trials with various values will determine the best value to obtain sufficient accuracy with maximum speed.

7.2 Simulation Time

All simulations start at zero time but the finishing time, TF, must be set by the user.

7.3 Number of Steps

A maximum permitted number of integration steps, NS, must be specified. The simulation will stop when more than NS steps have been taken.

7.4 Trace Facility

The trace facility is provided to allow the user to investigate values of variables at a specified time during a simulation. The desired trace time should be set using TR command. Simulation will stop when the trace time has been reached, and may be restarted as described in section 7.5.

Setting TR to zero inhibits the operation of the trace facility.

7.5 Commands

(a) SS start the simulation from zero time, using the initial conditions stored in the YSTART common block.

(b) RS restart the simulation from the current point in time.

7.6 Simulation Abort

Programmed simulation stops occur when time exceeds TR or TF or when NS steps have been taken.

Pressing sense switch A will cause an abnormal termination of a simulation. The RS command may be used to restart after this action.

7.7 Simulation Errors

There are three types of error message which may be typed out during the course of a simulation. These are detailed below.

(a) PAUSE NNNNN This is a standard FORTRAN error message and can arise for example by having a negative argument for the squareroot subprogram. Such error messages are fully detailed in the EAI 640 FORTRAN manuals.

(b) UNDEFINED CELL AT XXXXX This message indicates that a floating point arithmetic operation has been attempted using the contents of cell XXXXX before this cell has been defined. The relationship between this error message and the order in which the user's differential equations are written is described in section 3.4.

(c) UNDERFLOW AT XXXXX
OVERFLOW AT XXXXX
DIVIDE BY 0 AT XXXXX

These messages indicate the error conditons of exponent underflow, exponent overflow and attempted divide by zero respectively, arising in the floating point arithmetic.

The address XXXXX refers to the octal address of the call to the particular arithmetic routine which detected the error condition. XXXXX will usually be an address within the user's differential equation subroutine, which is loaded at '21537, but if XXXXX is lower down in core than this address, the error will have occurred in the integration program. In either case the error will be a consequence of the model parameter settings and possibly the value of the step length DT.

After one of the error messages in (b) or (c) above, computation will continue but the values will necessarily be inaccurate. Underflow errors occurring during simulation may be masked out by depressing SSW B. This will cause all underflows to be set to zero and processing will continue.

8.0 EXPERIMENTAL DATA

A special storage array has been set aside for the storage of experimental data which may be compared with simulated results.

A maximum of 50 experimental points may be read into this array from papertape.

The data should be prepared as follows:-

(a) Each value should be punched on one line as a real number in the format described in section 4.1.2.

(b) Each point is described by two numbers; the independent and dependent variable values.

(c) The independent variable value should be given first followed by the dependent variable value for each point in turn.

(d) The end of the tape must be marked with a "FORM" character.

(e) A "$" symbol at the start of a line will cause the contents of that line to be typed on the teletype, thus enabling data set identification. A comments line is not processed as a number.

(f) A "RUBOUT" character causes the previous character on that line to be deleted.

It is intended that such tapes be prepared and edited using the system program TED.

8.1 Data from previous simulations

As mentioned in section 6.0 it is possible to read in tapes which are punched using the LD command to device 5. This data, which occurs at equally spaced intervals starting at zero time can be read into either the save array or the experimental data array.

N.B. There is only room for 50 points in the experimental data array.

8.2 Commands

After typing the command ED the user is asked for the array (E or S) which the data is to be stored. Once this has been specified reading will commence.

N.B. Data which has unequal independent variable intervals will be stored incorrectly in the save array.

8.3 Errors on Tape

An error detected in the input field for a real number on tape will cause an error message to be typed which indicates which coordinate point pair contains the error. No number will be stored and reading will continue. No editing facilities are provided.

9.0 MEAN SQUARED ERROR

Executing the ME command causes the root mean squared error between the experimental data and one of the prepared or saved variables to be calculated. The user is asked to specify which variable is to be used. He should respond with the variable number in the case of a prepared variable or with the letter S if the saved variable is to be used. The calculation is performed as shown in the following formula.

$$M.S.E. = \left(\frac{\sum_{i=1}^{N} (A_i - B_i)^2}{N} \right)^{\frac{1}{2}}$$

where N = is the number of points in the experimental data array.

A_i is the ith experimental data point.

B_i is the value of the prepared or saved sample whose time value corresponds closest to that of A_i.

Experimental data points should not have negative time values for use in this calculation and any data point outside the range of the simulation samples on the time axis will be ignored.

10.0 AUXILLIARY COMMANDS

10.1 Monitor Return

The command MO causes control to be returned to the EAI DOS Monitor.

10.2 Comments

10.2.1 Teletype

The command CT allows the user to type any comments on the teletype. A "RUBOUT" should be typed as a terminating character.

10.2.2 Display

The command CD allows the user to type any comments on the display screen. After CD is typed the user should position the "write thru" cursor on the screen using the joystick. When this is in the right position the comment should be typed onto the keyboard. The terminating character is a "RUBOUT".

10.3 Changing the Model

The command CM is used when the user wishes to change his DIFEQN subroutine and then continue the simulation.

The command causes all of the common blocks of SIMUL8 to be

stored on a disc file named SIMCOM and a flag is set in cell '77661 of core to indicate that this has been done. Control is then handed over to the D.O.S. monitor or to the simulation operating system depending on which was in control when the simulation was set up. The simulation operating system is described in section 12.0.

As described in section 11.1, when SIMUL8 is reloaded and executed it will automatically load the contents of SIMCOM if it finds the flag in cell '77661 set. After loading, the flag is immediately reset.

10.4 Exchange Initial Condition and Solution Vectors

The command EY will cause the first NE cells of the YSTART common block and the EQTNS common block to be exchanged.

This command may be used to set up the initial conditions common with the steady state values of the variables. The YØ common should be initially loaded with arbitrary initial conditions and a simulation performed. If the plots show that all variables have reached steady state the EY command should be executed, thus transferring the steady state values into the initial condition common block. If steady state has not been reached the final values should be transferred into the initial condition common and the system simulated again. This should be repeated until a steady state is reached. It may be necessary in some cases to increase the final time TF so that the steady state can be reached quickly.

10.5 User Utility Program

This program may be executed at any time by using the UU command. The subroutine should be named UUPROG and should have no arguments. Due to core size restrictions this program should not contain any READ, WRITE, TYPE or ACCEPT statements.

If the user wishes to access the standard utilities available in SIMUL8, from within UUPROG he may do so by executing CALL's to the user modules described in Appendix VI. These utilities are stored on disc in the file SIMFCT and are loaded automatically by the operating system if required.

11.0 PROCEDURE FOR USE

In section 12.0 is described an operating system which carries out the editing, compiling and loading phases of setting up a user's DIFEQN program automatically. However, the following

instructions detail how to set up a simulation assuming that the user has an object file or tape of his DIFEQN subroutine previously prepared.

(a) Load the core-image file SIMUL8 using the D.O.S. monitor.

(b) Execute at '1000, a heading will be typed.

(c) Type in up to 30 characters which identify the simulation. This identification will appear as comments on any punched tape produced by SIMUL8.

(d) The message IP will be typed.

(e) 1. Type L,4 if the DIFEQN program object is on papertape.
2. Type L,NAMEXX if the DIFEQN program is in the file NAMEXX on disc.

(f) If all external calls are not resolved the message IP will be repeated and the user should then load the required object files or tapes.

(g) When all external calls have been satisfied the following information is then asked for:

1. Maximum number of simulation steps.
2. Integration step length.
3. Final simulation time.
4. Variables to be prepared for plotting (Max. 4)
5. Number of differential equations.
6. Values of the initial conditions. (Opportunity is given for these to come from papertape). See Appendix VI.
7. Values of the model parameters. (These may also be input from papertape). If model parameters are coming from the teletype the X character must be typed when all parameters have been input.

(h) X← is typed to indicate that SIMUL8 is ready to accept commands.

11.1 After Changing Model

If SIMUL8 is reloaded into core subsequent to a CM command (see section 10.3) having been used to dump the common blocks onto disc, a flag which this command sets in cell '77661 will be interrogated. If the program finds this flag set, steps (c) and (g) in section 11.0 will not be carried out. The common blocks which were saved will be loaded into core automatically and following step (f) the message X← will be typed as SIMUL8 waits for command input.

11.2 Re-entry

Re-entry of SIMUL8 at '1000 will cause the message X ← to be typed immediately without going through the normal setup procedure. It is not possible to load a new DIFEQN program without first re-loading SIMUL8 into core.

12.0 SIMUL8 OPERATING SYSTEM

A set of disc files which includes the file SIMUL8 has been arranged to manipulate the system processors FORTR and TED in such a way that changes to a model may readily be made at the source level. The system is based around the load and go feature of the Control Option Processor (COP). Four files exist containing COP control records which will load and execute the two processors and the file SIMUL8 in the appropriate order. These files will enable the procedures involved in text editing the source and compilation of the new source to be automatic. The newly created object will then be loaded into the simulation system automatically.

12.1 Concept and Use of the Operating System

The source statements of the user's model may be manipulated by the text editor between three source devices. These comprise the papertape station and two fixed source files on the disc. Compilation of the source statements from any of the source media is performed by FORTR and the resulting object code is output to an arbitrary disc file whose name is in the series OBJTAA, OBJTAB,.....OBJTAZ. SIMUL8 is then loaded and the last object file created is loaded automatically by the program loader within the package. The loading and executing of these processors and SIMUL8 is controlled by COP control records.

These COP control files contain COP commands to perform in the correct sequence the loading and execution of the text editor, Fortran compiler and SIMUL8. All source and object files are positioned and named automatically. Fortran compiler options e.g. MAP and LIST are selected by the user by loading an appropriate option file prior to execution of the selected COP control file. These options will apply for all subsequent compilations until an alternative option file is loaded. (See section 12.4).

The procedure for use is as follows:-

(a) Load the appropriate option file to set compiler options, using the DOS Monitor. Unit number 21 should be used.

(b) Position unit number 23 to the appropriate COP control file using the DOS Monitor.

(c) Transfer control to COP specifying unit number 23 as its input medium.

To carry out this sequence of operations the user should enter the following commands to the DOS Monitor.

#L, Optionfilename, 21

#P, Controlfilename, 23

#X, 23

Where Optionfilename is the name of the file containing the desired compiler options (see section 12.4), and Controlfilename is the name of the control file selected from section 12.2.

The example problem in Appendix III demonstrates the use of the operating system.

12.2 COP Control Files

There are four COP control files each of which performs similar tasks, the difference between the four being defined by the source media involved and whether editing is desired. The function of the four files is described below.

12.2.1 File 1 SIMPTC

This file will load and execute the Fortran compiler and initiate it to take source from papertape. SIMUL8 will then be loaded and the object code produced by the compiler automatically loaded. The user may then manipulate the model as described in the previous sections. On the execution of the CM (change model) command, control will be transferred to the COP file SIMPTT.

12.2.2 File 2 SIMPTT

This file will load and execute the text editor and set its device options to input text from papertape and output text to the first of the two disc files for source, SIMSC1. On exit from the text editor the updated source will be compiled and control transferred to SIMUL8 as described in 12.2.1. On executing the CM command, control will be transferred to the COP file SIMS1T.

12.2.3 SIMS1T

This file will allow editing of the contents of source file 1 (SIMSC1). Output from the editor will be written into the second source file (SIMSC2) which will be compiled and control transferred to SIMUL8 as described in 12.2.1. A CM command will transfer control to the COP control file SIMS2T.

12.2.4 SIMS2T

This file will allow the editing of the contents of SIMSC2. Output from the editor will be written into the source file SIMSC1 which will be compiled and control transferred to SIMUL8 as described in 12.2.1.A CM Command will transfer control to the COP file SIMS1T.

12.3 Editing the Source

Use of the text editing program (TED) is the same as described in the D.O.S. System Manual with the following three exceptions.

1. The F command has been invalidated.

2. The device options have been predefined by the COP control records.

3. The M command will return control to COP.

Source statements should be read from the primary source medium using the R or B commands, edited as usual and written into the secondary medium using the W, P or N commands. On completion of the editing process the M command should be used to return control to COP.

The user should take care to ensure that all of the source statements have been written into the secondary source medium before returning to COP via the M command. Thus if the W command is used, an E command should always be used before return to COP.

N.B. The source program should be terminated with an @.
If there is more than one source program (i.e. a DIFEQN routine, a UUPROG, and one or more function block routines supplied by the user) then the @ should follow the last subroutine.

12.4 Fortran Compilation

The source input and object output mediums are controlled by the COP control records and need not concern the user. The user has control over the printing of listings and compiler maps with the use of COP option files. These may be created by the user by following the procedure laid down in the DOS manual. Alternatively the user may use one of four COP option files supplied with the operating system. These files reside on the disc and may be loaded with the DOS monitor prior to entering the operating system. The four option files are described below.

12.4.1 Option File 1. SIMLIS

This option file will cause the compiler map to be suppressed and the compiler source list to be output to the teletype.

12.4.2 Option File 2. SIMMAP

This option file will suppress the compiler listing and output the compiler map to the teletype. Errors will also be output to the teletype.

12.4.3 Option File 3. SIMNLM

This option file will suppress both compiler source and compiler map but will list the compiler errors on the teletype.

12.4.4 Option File 4. SIMLM

This option file will allow the compiler source listing and the compiler map to be output to the teletype.

N.B. It is essential that one of the COP option files is loaded into core before control is transferred to the Simulation Operating System.

In general, the option file SIMNLM should be used. The user may obtain his source listings using the hard copy unit during the editing phase. The option files that give compiler maps are only required when "UNDEFINED CELL" error messages are being investigated. The option files that give source listings will be used when a tidy copy of the user's source statements is required for xeroxing.

12.5 Execution of SIMUL8

Under the operating system SIMUL8 is loaded automatically and a flag is set in the program to enable automatic control of the program loader. After execution of SIMUL8 the name of the last generated object is fed into the program loader and this file is loaded automatically. If the program loader is not satisfied by this program the standard function block program name will be fed in followed if necessary by the name RTL. If neither of these programs satisfy the program loader then control will be returned to the keyboard to allow the user to load the externs in the usual way.

If the user wishes to change the source statements of the model during the simulation then he should type the CM command to the command decoder. This will cause the following sequence of events to occur.

1. The common blocks will be written into the file SIMCON.
2. Control will be transferred to the COP file that initially loaded the simulation.
3. This COP file will transfer control to the next COP control file which will allow the processes of text editing and compilation to proceed.
4. SIMUL8 will be loaded into core and the object files loaded as described before.
5. The common areas saved in SIMCOM will be reloaded and control handed over to the command decoder. This procedure may be repeated as many times as desired, each time the appropriate COP file will be given control and the editor and compiler handled automatically.

12.6 Obtaining a Papertape Copy of Model Source Statements

A papertape copy of the model source statements may be obtained from the text editor either during the editing phase of the simulation or by loading and executing TED after the simulation has been terminated.

12.6.1 Papertape Copy During Editing Phase

After writing the source statements onto the secondary source medium using the W command, the user should change the text-output device to 5 using the X command. The source statements may then be output to the punch using the W or P commands. Care should be taken when the source statements occupy more than one page. It is advised that the entire source is read into the TED buffer, and written onto the secondary source medium before the output device is changed to the punch.

Below is a list of the devices used in TED for the three COP control files that allow editing.

1. SIMPTT 4,22,
2. SIMS1T 21,22, (Input device, Output device)
3. SIMS2T 21,22,

12.6.2 Papertape Copy After Leaving the Simulation System

The user should find the name of the most recent source file by observing the name of the source file immediately above the last point of execution of FORTR. This file may then be dumped onto papertape using TED in the usual way.

N.B. The user is advised against using DOS FUTIL DUMP to obtain a hard copy of the source statements. This program does not punch the data in a format which is suitable for the Fortran compiler. Also DUMP will dump the whole file which could be considerably longer than the actual source statements in the file.

12.7 Returning to the Monitor

While the simulation is under the control of the Operating System a flag is set in cell '37722 to say that COP is in control. Executing the DOS BOOTSTRAP at any time will reload COP and cause if to continue reading records from the last used COP control device.

To exit to the DOS monitor at any time during the execution of SIMUL8 the user should use the MO command.

At any other time when exit to the DOS monitor is desired the user should clear the COP flag by clearing core, and then execute the DOS BOOTSTRAP.

12.8 Errors While Using the Simulation System

12.8.1 Compiler Errors

Any errors detected by the compiler will be listed on the teletype and a Fortran error number assigned. After the entire program has been scanned the message "JOB ERROR" will be typed and control will return to the DOS MONITOR.

If an error is found the user should position to the next COP control file and transfer control back to COP using the command X,23. Section 12.2 holds the information necessary to determine the name of the next COP control file.

12.8.2 System Errors

If an error occurs during the execution of COP an error number will be typed followed by the message "SYSTEM FAILURE" and control will be returned to the monitor. If the error is an error 4 it is likely that the COP option parameters have been destroyed or have not been loaded. The user should reload the appropriate COP option file and restart the operating system.

Other errors may be caused by parts of the Operating system on the disc being destroyed. This should be checked by running MAPIT. If this appears to be O.K. then the user should restart the operating system.

12.9 Summary of the Files in the SIMUL8 Operating System

SIMUL8	Simulation system mainline.
SIMSC1	First source file.
SIMSC2	Second source file.
SIMCOM	Common block storage file.
SIMPTC	COP control file to take a papertape source, compile it and transfer the object produced to SIMUL8.
SIMPTT	COP control file to take a papertape source, edit and compile it and transfer the object produced to SIMUL8.
SIMS1T	COP control file to accept source from SIMSC1, edit and compile it and transfer the object produced to SIMUL8.
SIMS2T	COP control file to accept source from SIMSC2, edit and compile it and transfer the object produced to SIMUL8.

SIMLIS	COP option file to give compiler listing and to suppress the compiler map.
SIMMAP	COP option file to give a compiler map and error listing but to suppress the source listing.
SIMNLM	COP option file to give error listings but to suppress the compiler source list and map.
SIMLM	COP option file to give compiler source list and a compiler map.
SIMFCT	Library file containing object coding of standard simulation system function blocks.

Appendix I

Details of COMMON Areas

COMMON MODPAR Starting Address '26221

Interrogation Command	Element Number	Description
MP	1 ' ' ' ' 60	Storage for model parameters which the user wants to control.

COMMON YSTART Starting Address '26125

Interrogation Command	Element Number	Description
YØ	1 ' ' ' ' 30	Storage for initial conditions of differential equation solutions.

COMMON DERIV Starting Address '26031

Interrogation Command	Element Number	Description
None	1 ' ' ' 30	Storage of derivatives of differential equation solutions.

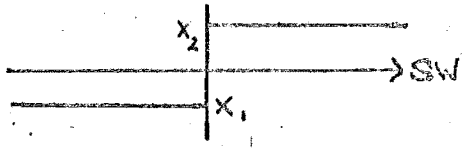
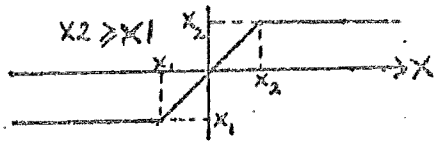
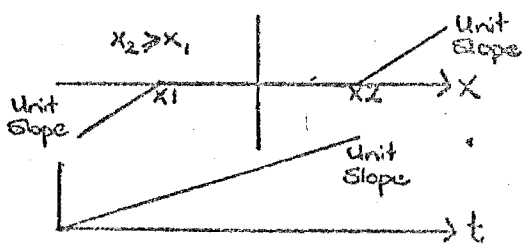
COMMON EQTNS Starting Address '25641

Interrogation Command	Element Number	Description
YY	1 ' ' ' ' ' ' 60	First NE elements have differential equation solutions in order. Other elements may be used for quantities required for output. (see 3.1 and 5.1)

Appendix II

Standard Function Blocks

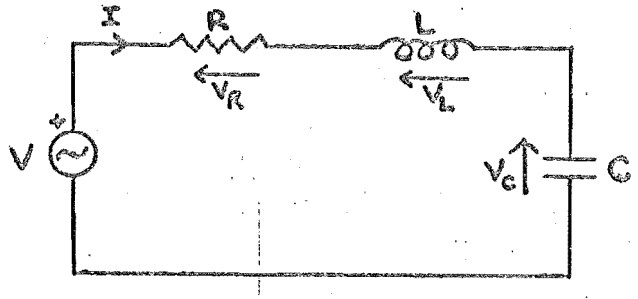
The following is a list of the standard function blocks available with SIM. These subprograms are available for use in the users DIFEQN subroutine as FUNCTION subprograms. They are contained in a separate disc file and must be loaded after DIFEQN has been loaded (see section 11.0). In addition to these special functions all standard FORTRAN functions such as SIN, COS, EXP, LOG are available to the user.

Function Name	Function Call	Description
Pulse	PULSE (T1,T2)	
Step	STEP(T1)	
Ramp	RAMP(T1)	
Negative Clip	CLIPNG(X)	
Positive Clip	CLIPPS (X)	
Relay	RELAY(X1,X2,SW)	
Limiter	ZLIMIT(X1,X,X2)	
Bang-bang	BANG(X)	
Dead Space	DEADSP(X1,X,X2)	
Time Generator	TIME(DUMMY)	

Appendix III

Example Problem

As an example problem we use SIMUL8 under control of the operating system to investigate the response of a series RLC circuit to step and sinusoidal inputs.



Using the elementary relationships between current and voltage in each circuit element we can set up the following equations:

$$V_R = IR$$

$$V_L = V - V_C - V_R$$

$$\frac{dI}{dt} = V_L/L$$

$$\frac{dV_C}{dt} = I/C$$

These equations were written into a DIFEQN subroutine according to the rules detailed in section 3.1. This program was punched onto paper tape using the off-line teletype. Listings are given on subsequent pages; the following points should be noted when considering these:

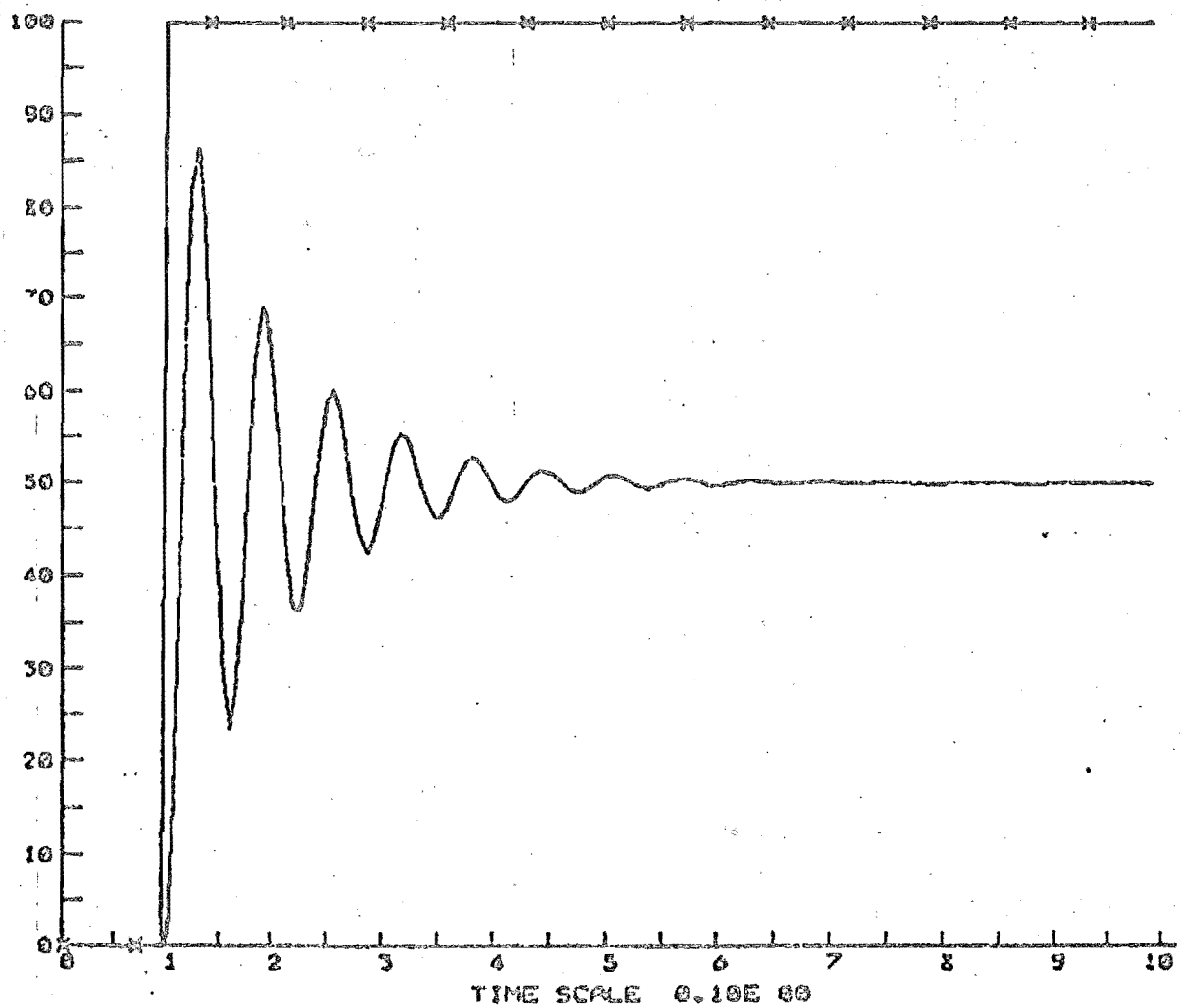
1. The variables I and L had to be declared REAL due to the Fortran naming convention.
2. Only the first 2 cells of EQTNS are used for differential equation solutions. The quantities V, VR and VL are in EQTNS because they may be required for output.
3. All constants which it is desired to control are stored in MODPAR.
4. The equations are ordered so that VR is calculated before VL, and VL is computed before DI.

On the following pages is given all the information recorded on the teletype during this study and 6 plotted graphs of the results.

```
#L,SIMNLM,21
SIMNLM CI
LD
#P,SIMPTT,23
SIMPTT SO
#X,23
.COP CI
LD
C←
    $JOB    PAPERTAPE-TED-SIMSCI-FORTR-SIMUL8
C←
SIMSCI SO
TED CI
LD
.COP CI
LD
C←
FR
SIMSCI SO
FR
OBJTAA OB
FORTR CI
LD

JOB OK

.COP CI
LD
C←
SIMUL8 CI
LD
```

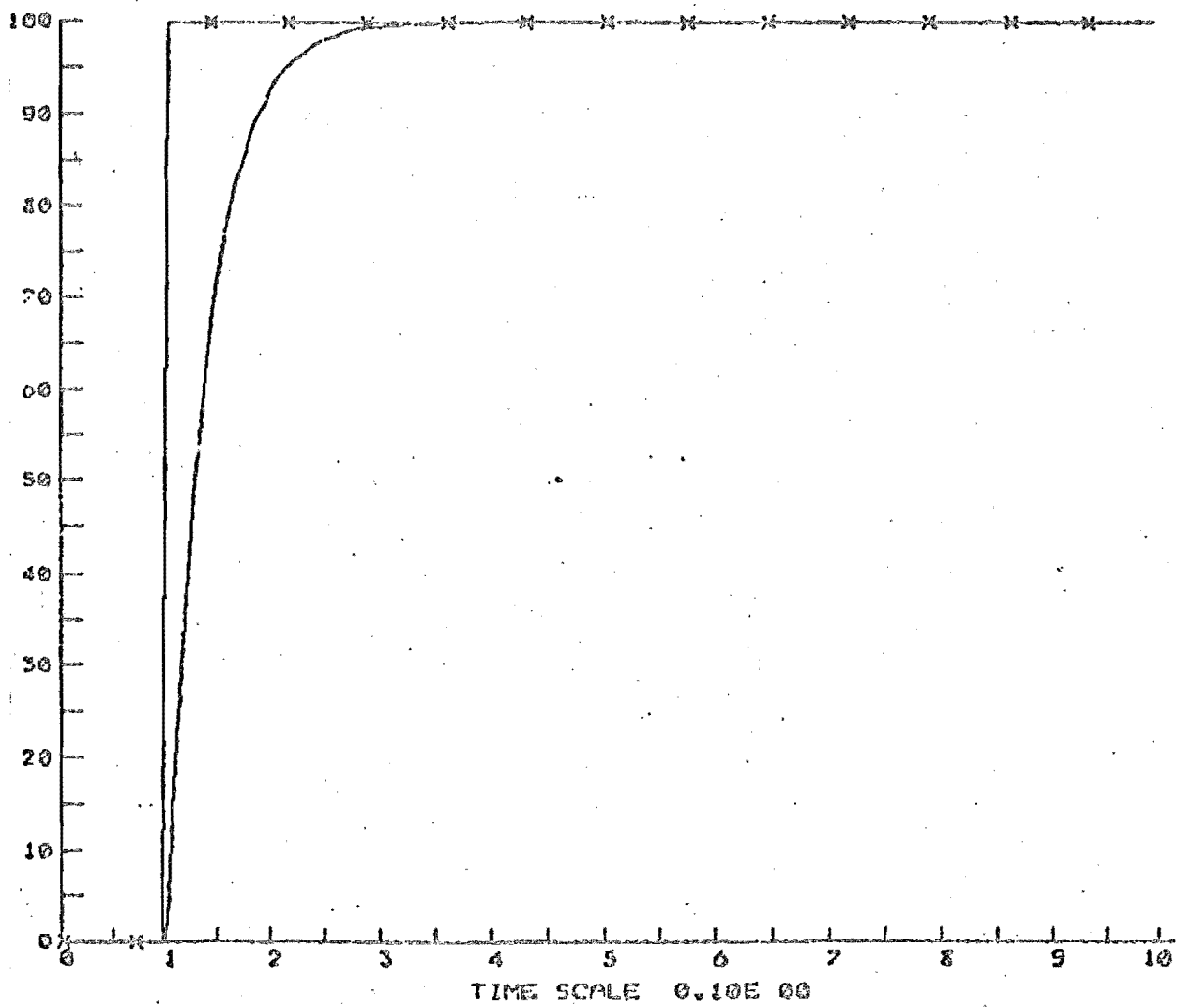
```
XE: TED
XX: COMMAND
XX:
XX: M
XX:
XX: L
001:  C
002:  C
003:  C       DEMONSTRATION DIFEQN SUBROUTINE
004:  C        SERIES RLC CIRCUIT WITH PULSE INPUT
005:  C
006:  C
007:          SUBROUTINE DIFEQN
008:          REAL I,L
009:          COMMON /EQTNS/ I,VC,V,VR,VL
010:          COMMON /DERIV/ DI,DVC
011:          COMMON /MODPAR/ R,L,C,TP1,TP2
012:  C
013:  C       ALGEBRAIC EQTNS
014:  C
015:          V=PULSE(TP1,TP2)
016:          VR=I*R
017:          VL=V-VR-VC
018:  C
019:  C       DIFFERENTIAL EQTNS
020:  C
021:          DI=VL/L
022:          DVC=I/C
023:          RETURN
024:          END
025:  @
026:
026:  P
026:
026:  M
```

```
PAGE  01


E.A.I. 640 DIGITAL SIMULATION


IDENTIFY← DEMONSTRATION  25 / 1 / 73
IP

LI
IP

NO. OF STEPS← 4000
STEP LENGTH← .001
FINAL TIME← 1.0
VARIABLES TO BE PREPARED ← 2,3,4,5
NO. OF DIFF. EQTNS.← 2
INITIAL COND. DEVICE← 2
 1← 0.
 2← 0.
MODEL PARAMETER DEVICE← 2
 1← 20.
 2← 1.0
 3← 1.0E-4
 4← 0.1
 5← 10.
 6← X
X←
SS RUN NO.  1
X←
DA ← 2,3
X←
CT THIS IS UNDERDAMPED RESPONSE. GET HARD COPY
   ALSO GET PAPERTAPE DUMP OF VARIABLE 2.
X←
LD  DEVICE NO. ← 5
 VARIABLE NO. ← 2
    NO. OF PTS. ← 200
X←
CT INCREASE RESISTANCE TO MAKE RESPONSE OVERDAMPED
X←
MP ← 1:  0.2000E 02   ← 400.
   ← X
X←
SS RUN NO.  2
X←
DA ← 2
X←
DO ← 3
X←
CT GET HARD COPY.
X←
CT READ IN PREVIOUSLY PUNCHED RESPONSE TO COMPARE
X←
ED  ARRAY E OR S ← S
```



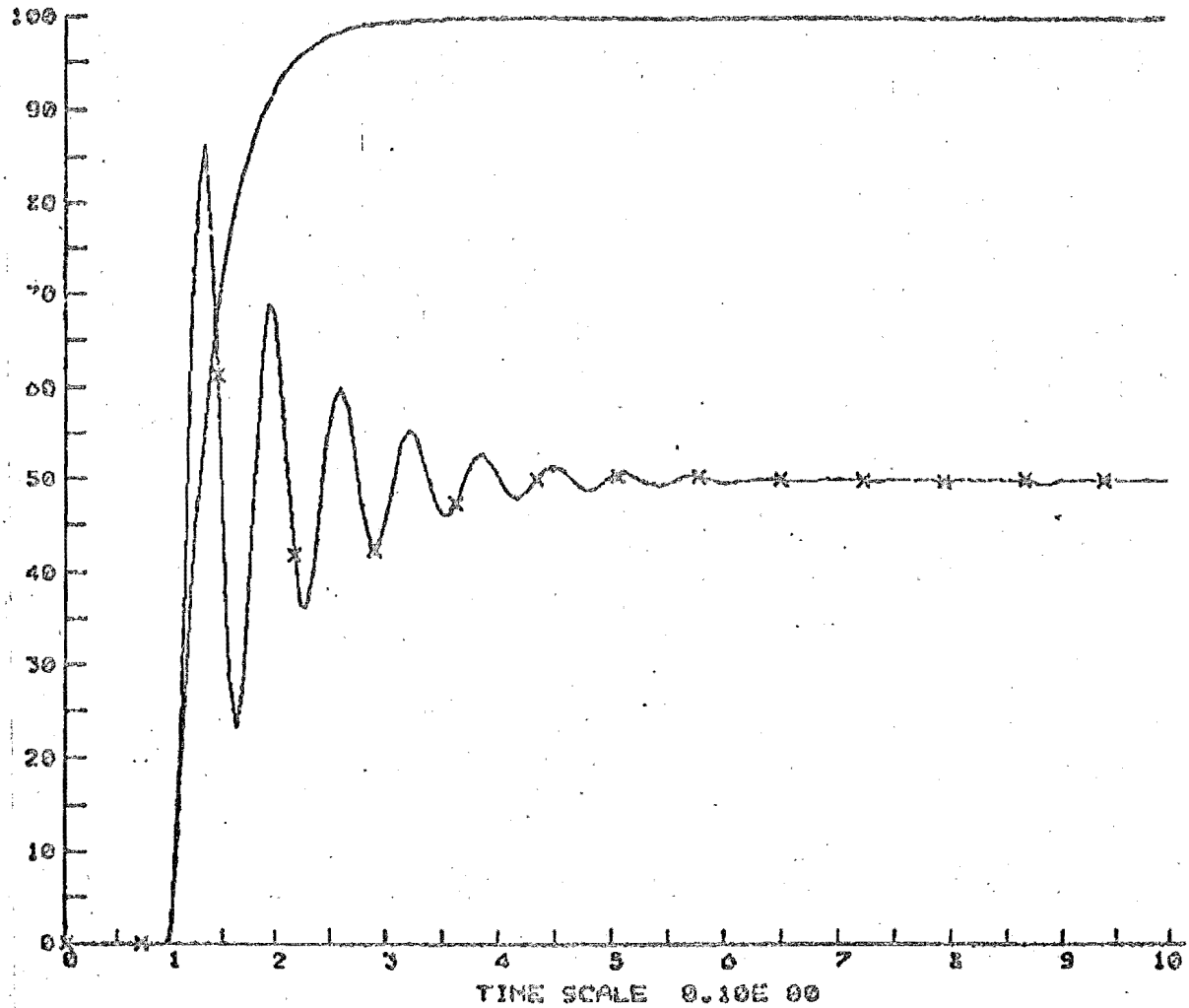
SYMBOL	VARIABLE	RUN NO.	SCALE	ORIGIN	MAXIMUM	MINIMUM
———	2	1	0.20E-01	0.0	0.173E 01	0.0
——×——	3	1	0.10E-01	0.0	0.100E 01	0.0

GRAPH 1 DEMONSTRATION 25 / 1 / 73



SYMBOL	VARIABLE	RUN NO.	SCALE	ORIGIN	MAXIMUM	MINIMUM
———	2	2	0.10E-01	0.0	0.999E 00	0.0
X———X	3	2	0.10E-01	0.0	0.100E 01	0.0

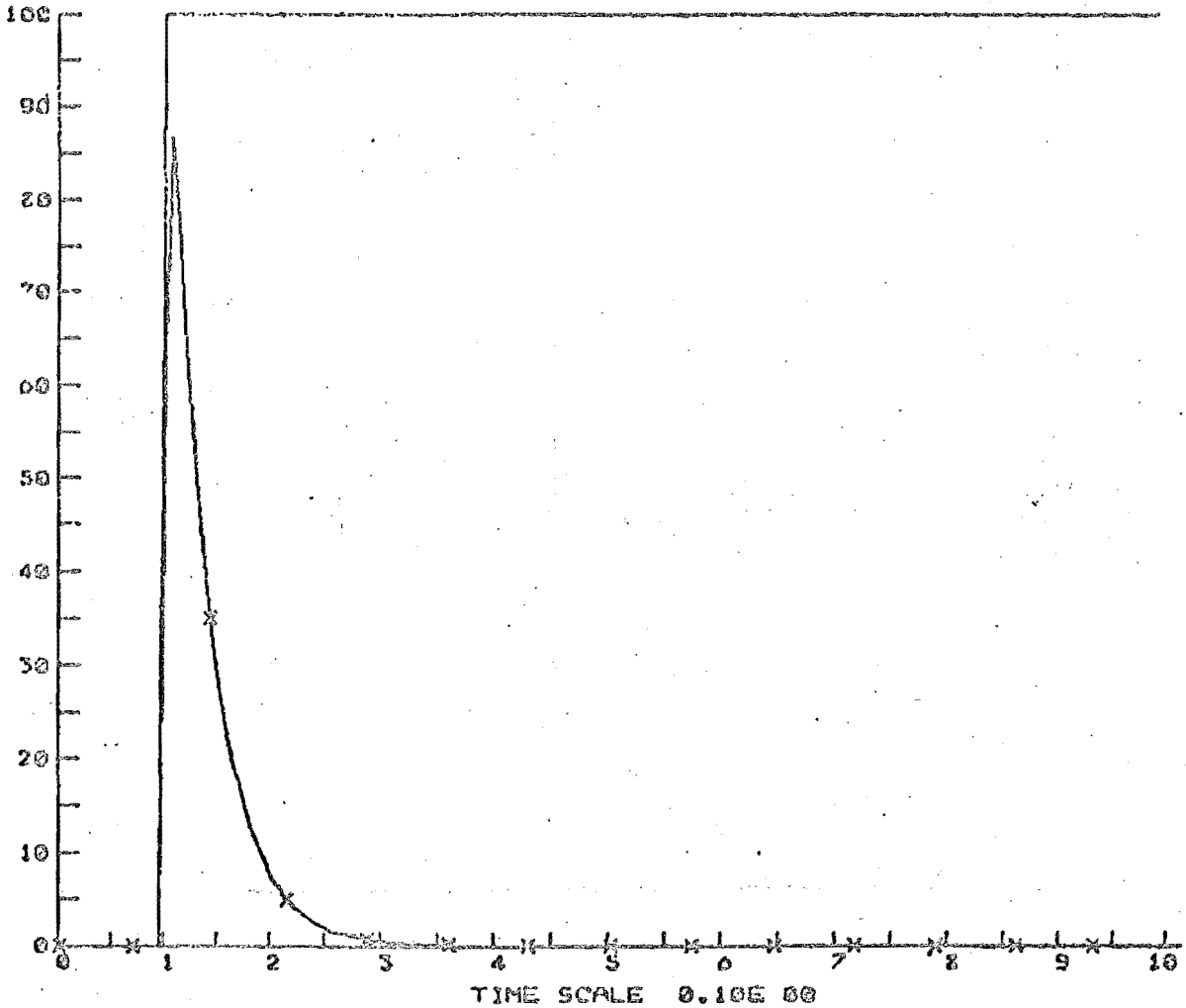
GRAPH 2 DEMONSTRATION 25 / 1 / 79



SYMBOL	VARIABLE	RUN NO.	SCALE	ORIGIN	MAXIMUM	MINIMUM
———	2	2	0.10E-01	0.0	0.999E 00	0.0
X——X	0	2	0.20E-01	0.0	0.173E 01	0.0

GRAPH 3 DEMONSTRATION 25 / 1 / 73

```
PAGE  02

  VARIABLE  2 FROM RUN  1

 DEMONSTRATION  25 / 1 / 73

X←
CT THIS COULD HAVE BEEN ACHIEVED BY COMMANDING SA ← 2
    BEFORE LAST SIMULATION INSTEAD OF USING PAPERTAPE
X←
DA ← 2,S
X←
CT HAVE A LOOK AT THE CURRENT. THIS WAS NOT PREPARED....
X←
PR  ← 1,3,4
X←
SS RUN NO.  3
X←
DA ← 3,1
X←
CT NOW CHANGE MODEL TO HAVE A SINUSOIDAL INPUT
X←
CM
   SIMCOM DA
.COP CI
LD
C←
SIMSIT SO
    $JOB    SIMSC1-TED-SIMSC2-FORTR-SIMUL8
C←
SIMSC1 SO
SIMSC2 SO
TED CI
LD
.COP CI
LD
C←
OBJTAA OB
FR
SIMSC2 SO
FR
OBJTAB OB
FORTR CI
LD

JOB OK

.COP CI
LD
C←
SIMUL8 CI
LD

IP

LI
IP

LI
IP

SIMCOM DA
```

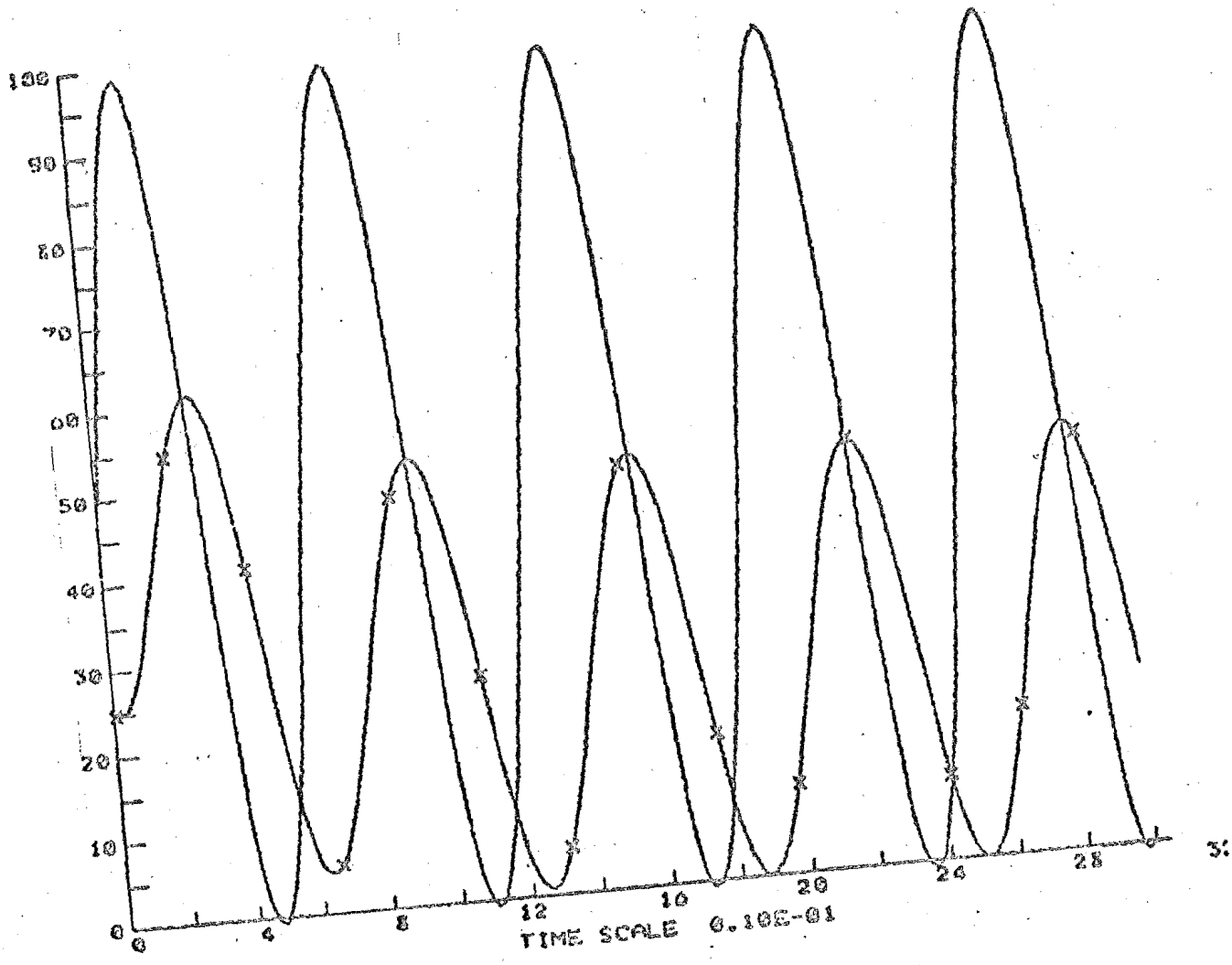


SYMBOL	VARIABLE	RUN NO.	SCALE	ORIGIN	MAXIMUM	MINIMUM
———	3	3	0.10E-01	0.0	0.100E 01	0.0
X——X	1	3	0.25E-04	0.0	0.217E-02	0.0

GRAPH 4 DEMONSTRATION 25 / 1 / 73

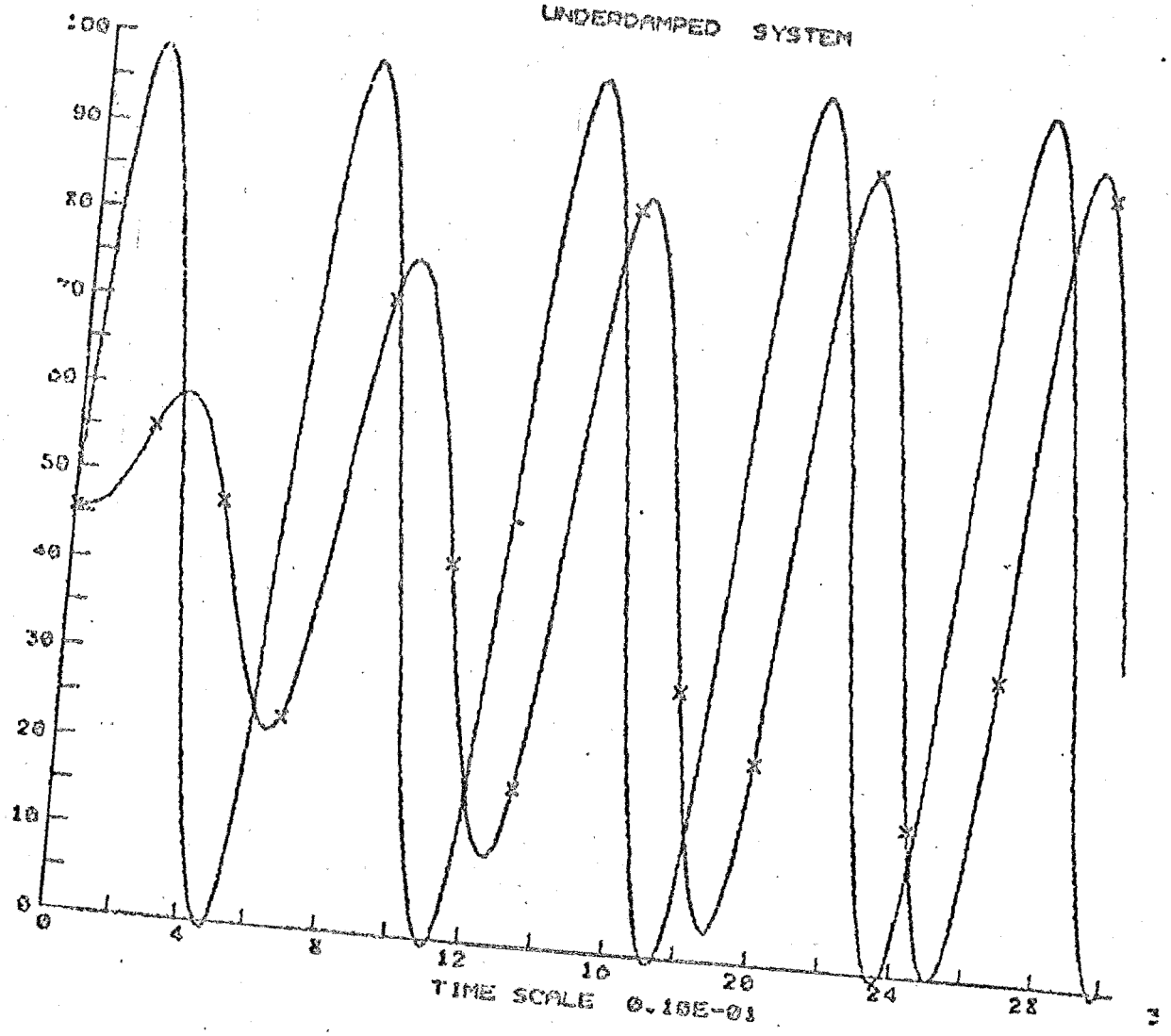
```
EE: COMMAND
EE:
EE: N
EE:
EE: 4.PULSE
004: C       SERIES RLC CIRCUIT WITH PULSE INPUT
TEXT                                 SINUSOID
005: COMMAND
005:
005: 15,C
015: TEXT
015:       V=AMPLIT*SIN(OMEGA*TIME(DUMMY))
01D: A
016: \
016: COMMAND
016:
016: 11,TP1,TP2
011:       COMMON /MODPAR/ R,L,C,TP1,TP2
TEXT                               AMPLIT,OMEGA
012: COMMAND
012:
012: L
001: C
002: C
003: C       DEMONSTRATION DIFEQN SUBROUTINE
004: C        SERIES RLC CIRCUIT WITH SINUSOID INPUT
005: C
006: C
007:       SUBROUTINE DIFEQN
008:       REAL I,L
009:       COMMON /EQTNS/ I,VC,V,VR,VL
010:       COMMON /DERIV/ DI,DVC
011:       COMMON /MODPAR/ R,L,C,AMPLIT,OMEGA
012: C
013: C       ALGEBRAIC EQTNS
014: C
015:       V=AMPLIT*SIN(OMEGA*TIME(DUMMY))
016:       VR=I*R
017:       VL=V-VR-VC
018: C
019: C       DIFFERENTIAL EQTNS
020: C
021:       DI=VL/L
022:       DVC=I/C
023:       RETURN
024:       END
025: @
026:
026: P
026:
026: N
```

```
PAGE  03


X←
MP ← 1,5:
     0.4000E 03  0.9999E 00  0.1000E-03  0.1000E 00  0.1000E 02

   ← 4:  0.1000E 00   ← 1.0
   ← 5:  0.1000E 02   ← 100.0
   ← X
X←
CT THIS WILL DRIVE AT NATURAL FREQUENCY
X←
SS RUN NO.  4
X←
DA ← 3,2
NOT PREPARED
X←
PR  ← 2,3,4,5
X←
SS RUN NO.  5
X←
DA ← 3,2
X←
CT TO HARD TO READ. TRY FEWER CYCLES.
X←
TF   0.9999E 00   ← 0.3
X←
SS RUN NO.  6
X←
DA ← 3,2
X←
CT GET HARD COPY.
X←
CT NOW GO BACK TO UNDERDAMPED PARAMETERS
X←
MP ← 1:  0.4000E 03   ← 20.0
   ← X
X←
SS RUN NO.  7
X←
DA ← 3,2
X←
CD
X←
CD
X←
CT GET HARD COPY
X←
CT DEMONSTRATION ENDS RETURN TO MONITOR
   WHAT SAY YOU HAVE A GO EH!
X←
MO M←
```



SYMBOL	VARIABLE	RUN NO.	SCALE	ORIGIN	MAXIMUM	MINIMUM
———	3	6	0.20E-01	0.49E 02	0.100E 01	-.100E 01
X——X	2	6	0.10E-01	0.25E 02	0.309E 00	-.250E 00

GRAPH 7 DEMONSTRATION 25 / 1 / 75



SYMBOL	VARIABLE	RUN NO.	SCALE	ORIGIN	MAXIMUM	MINIMUM
———	3	7	0.20E-01	0.49E 02	0.100E 01	-.100E 01
×——×	2	7	0.10E 00	0.40E 02	0.474E 01	-.401E 01

GRASH 8 DEMONSTRATION 25 / 1 / 73

Appendix IV

Input/Output Device Numbers

Number	Device
1	Teletype typer
2	Teletype keyboard
3	Display
4	Papertape reader
5	Papertape punch
21	Disc Source Input File
22	Disc Source Output File

Appendix V

Command Summary

Manipulation of Parameters (Teletype)(Sect. 4.1)

Command	Parameter(s) Interrogated	Remarks
NS	Max. number of integration steps (Sect. 7.3)	Value of parameter is immediately displayed and new value is waited for. CR causes current value to be retained.
NE	Number of differential eqtns.	
DT	Integration step length (Sect.7.1).	
TF	Simulation finish time (Sect.7.2).	
TR	Trace time (Sect. 7.4).	
YØ	Common YSTART (initial conds)	After accepting the command the program waits for the user to identify the element(s) to be interrogated.
YY	Common EQTNS (depend. vars)	
MP	Common MODPAR (Model parameters)	

Manipulation of parameters (Papertape)(Sect. 4.2)

Command	Effect	Remarks
RY	Read initial conditions from papertape.	Tape format given in section 4.4.2.
RP	Read model parameters from papertape.	
OY	Punch NE initial conditions onto papertape.	
OP	Punch model parameters onto papertape.	Number of values to be output is asked for.

Simulation control (Sect. 7.5)

Command	Effect	Remarks
SS	Start simulation from zero time.	Simulation stops when more than NS steps have been taken, when time exceeds TR or TF or SSW A is pressed.
RS	Restart simulation after an interruption.	

Plotting (Sect. 5.2)

Command	Effect	Remarks
DA (Automatic)	Erase screen, draw grid and plot each of the variables with automatic scaling and origin suppression.	

Plotting (Sect. 5.2)(cont.)

Command	Effect	Remarks
DF (Family)	Erase screen, draw grid and plot each of the variables with the scaling and origin suppression of the first variable.	1. After accepting the command SIMUL8 wait for the user to input the numbers of the variables or arrays to be plotted. 2. SSW C inhibits origin suppression.
DO (Old)	Plot each of the variables on the current grid using the scaling and origin of the preceding curve drawn.	
DN (New)	Plot the variables on the current grid using automatic scaling and origin suppression for each.	

Dumping and Reading Data (Sect. 8.0)

Command	Effect	Remarks
ED	Read data from papertape into the save array or the experimental data array.	See Sect. 8.0 for tape format.
LD	List one of the data storage array contents onto teletype, display or paper-tape punch.	

Miscellaneous

Command	Effect	Remarks
CT	Allows comments to be typed on the teletype (Sect. 10.2.1).	"Rubout" is the terminating character.
CD	Allows comments to be typed on the display screen. (Sect. 10.2.2).	"Rubout" is the terminating character. Position text using joystick.
CM	Dump common areas onto disc. Set flag in cell '77661 and execute the DOS Monitor. (Sect. 10.3 and 12.0).	If the SIMUL8 Operating System is in use control will be transferred to it.
ME	Calculate RMS error between the experimental data array and another specified array. (Sect. 9.0).	User should enter variable number or the letter S for the save array.
MO	Return to DOS Monitor (Sect. 10.1 and 12.7).	

Miscellaneous (cont.)

Command	Effect	Remarks
EY	Exchange NE cells of YSTART with NE cells of EQTNS common. (Sect. 10.4).	
SA	Save the samples of a prepared variable for later inspection. (Sect. 5.1.2).	User should enter variable number.
UU	Execute the User Utility program UUPROG. (Sect. 10.5)	
PR	Prepare variables for plotting during subsequent simulations. (Sect. 5.1.1).	Maximum of four variable numbers to be entered.

APPENDIX VI

Access to SIMUL8 commands from UUPRO6

Usage	Function	Equivalent to
CALL UMP(I,VAL)	Sets model parameter No. to value VAL.	MP I: OLDVAL ← VAL
CALL UYØ (I,VAL)	Sets initial condition No. I to value VAL.	YØ I: OLDVAL ← VAL
CALL USS	Solves the model equations	SS
CALL UDA(I) *	Draws a new grid and plots variable No. I with automatic scaling	DA I
CALL UDO (I) *	Plots variable No. I on the previously drawn grid using the scale factors of the last plotted variable.	DO I
CALL UDN (I) *	Plots variable No. I on the previously drawn grid with automatic scaling.	DN I

* When using the plot modules I = 0 causes the experimental data away to be plotted and I = 1 the save away.

References

Akima H. (1970). A new method of interpolation and smooth curve fitting based on local procedures. J.A.C.M. 17:589

Altman P.L., & Dittmer D.S. (1961). Blood and other body fluids. Biological handbook. Fed. of Amer. Soc. for Exp. Biol.

Bashforth F., & Adams J.J. (1883). Theories of Capillary action. Cambridge Univ. Press.

Beaven D.W., Espiner E.A., & Hart D.S. (1964). The suppression of cortisol secretion by steroids, and response to corticotrophin in sheep with adrenal transplants. J. Physiol. 171:216.

Bennett W.R. (1956). Methods of solving Noise Problems. Proc. I.R.E. p. 609.

Berger G.M.B. (1971). Possible role of cyclic AMP in the short term regulation of adrenal steroidogenesis. Nature 232:474.

Berman M., & Weiss M.F. SAAM - Simulation, Analysis And Modelling. National Inst. of Health, Bethesda, Maryland U.S. Govt. Printing Office Washington D.C. 20402.

Bessar G.M., Orth D.N., Nicholson W.E., Byyny R.L., Abe K., & Woodham J.P. (1971). Dissociation of the disappearance of Bioactive and Radioimmunoreactive ACTH from Plasma in Man. J. Clin. Endocr. 32:595.

Bethune J.E., Nelson D.H., & Thorn G.W. (1957). Plasma adrenocorticotrophic hormone in Addison's disease and its modification by the administration of adrenal steroids. J. Clin. Invest. 36:1701

Birmingham N.K., & Kurlents E. (1968). Inactivation of ACTH by isolated rat adrenals and inhibition of corticoid formation by adrenocortical hormones. Endocr. 62:47.

Borkowski A., Delcroix C., & Levin S. (1972a). Metabolism of adrenal cholesterol in man. I. In vivo studies. J. Clin. Invest. 51:1664.

Borkowski A., Delcroix C., & Levin S. (1972b). Metabolism of adrenal cholesterol in man. II. in vitro studies including a comparison of adrenal cholesterol synthesis with the synthesis of the glucocorticoid hormones. J. Clin. Invest. 51:1679.

Brennan R.D., & Silberberg M.Y. (1968). The system/360 continous system modelling program. Simulation 11:301.

Burstein S., & Gut M. (1971). Biosynthesis of pregnenolone. Recent Prog. in Horm. Res. 27:303.

Cats A., & Kassenaar A.A.H. (1957). Influence of the kidney on the disappearance rate of labelled corticotrophin from the blood stream. Acta Endocr. 24:43.

Clegg A.G., & Clegg P.C. (1969). Hormones cells and organisms. Heinnemann Press.

Dallman M.F., & Yates F.E. (1969). Dynamic asymmetries in the corticosteroid feedback path and distribution-metabolism-binding elements of the adrenocortical system. Ann. N.Y. Acad. Sci. 156:696.

Daniels F., & Alberty R.A. (1966). Physical chemistry III Edition. Wiley International (Publ.)

Dolkas C.B., & Leon H.A. (1970). A nonlinear time varying mathematical anolog of the gluco-corticoid control. I.E.E.E. Trans. Biomed Eng. BME 17:1

Donald R.A. (1968). Application of the coated charcoal separation method to the radio-immunoassay of plasma corticotrophin. J. Endocr. 41:499.

EAI (1970). A.P.S.E. (Automatic Programming and Scaling of Equations) User Manual (Fortran Version). Manchester Nat. Computing Centre Ltd.

Elgerd O.L. (1967). Control System Theory. McGraw & Hill (Publ.)

Espiner E.A., Donald R.A., Hart D.S., Ross Janne, & Jordan R.B. (1974). Evidence for the Adrenocortical uptake of ACTH in vivo. Am. Jnl Physiol. 226:96.

Espiner E.A., Jensen C.A., & Hart D.S. (1972a). Dynamics of adrenal response to sustained local ACTH infusions in conscious sheep. Am. J. Physiol. 222:570.

Espiner E.A., Hart D.S., & Beavan D.W. (1972b). Cortisol secretion during acute stress and response to dexamethasone in sheep with adrenal transplants. Endocr. 90:1510.

Farese R.V. (1971). Stimulation of pregnenolone synthesis by ACTH in rat adrenal sections. Endocr. 89:958.

Fortier C., & De Groot J. (1959). Adenohypophysial corticotrophin and plasma free corticosterioids during regeneration of the enucleated rat adrenal gland. Am. J. Physiol. 196:589.

Frieden E., & Lipner H. (1971). Biochemical endocrinology of the vertebrates. Foundations of modern biochemistry series. Prentice-Hall.

Fromm N.C. (1970). EAI Software Report No. 70-10SW.

Gann D.S. (1969). Unidirectional rate sensitivity in cardiovascular and endocrine functions. Part II. Parameters of the stimulus initiating the adrenocortical response to hemorrhage. Annals N.Y. Acad. Sci. 156: 740.

Gann D.S., Ostrander L.E., & Schoeffler F.D. (1968). A finite state model for the control of adrenal cortical steroid secretion. In: Systems theory and biology M.D. Mesarovic' (Ed.) Springer Verlag. P.185

Gear C.W. (1971). The automatic integration of ordinary differential equations. Comm. A.C.M. 14:176.

Gill G.N. (1972). Mechanism of ACTH action. Metabolism 21:571.

Glasstone S., & Lewis D. (1963). Elements of Physical Chemistry. McMillan and Co. Ltd (Publ.)

Hales C.N. (1972). Immunological techniques in diabetes research. Diabetologia 8:229.

Hall P.F., & Young D.G. (1968). Site of action of trophic hormones upon the biosynthetic pathways to steroid hormones. Endocr. 82:559.

Haynes R.C. (1958). The activation of adrenal phospharylase by the adrenocorticotrophic hormone. J. Biol. Chem. 233:1220.

Haynes R.C., & Berthet L. (1957). Studies on the mechanism of action of adrenocorticotrophic hormone. J. Biol. Chem. 225:115.

Haynes R.C., Koritz S.B., & Peron F.G. (1959). Influence of adenosine 3'5' monophosphate on corticoid production by rat adrenal glands. J. Biol. Chem. 234:1421.

Henrici P. (1962). Discrete variable methods in ordinary differential equations. Wiley. New York.

Herscovitch H., & Schneider T. (1965). GPSS-III - An expanded general purpose simulator. IBM systems Jnl. 4:174.

Hodges J.R., & Vernikos J. (1960). The effects of hydrocortisone on the level of corticotrophin in the blood and pituitary glands of adrenalectomized, and of stressed adrenalectomized rats. J. Physiol. 150:683.

Hooke R., & Jeeves T.A. (1961). "Direct search" solution of Numerical and statistical problems. Jnl. Assoc. Comp. Mach. 8:212.

IBM (1966). Program reference manual-1130 Continuous system modelling program. IBM Corporation.

Ingle D. (1959). Current status of adrenocortical research. Am. Scientist 47:413

Jones M.T., & Neame R.L.B. (1971). Evidence in favour of a fast feed-back control of ACTH secretion by corticosterone. Jnl Physiol. 216:74P.

Jordan R.B. (1969). Magnetic Tape Store and Interface - The Interfacing Hardware. Third Prof. project report. Electrical Engineering Dept. Univ. of Cant. Christchurch N.Z.

Karaboyas G.C., & Koritz S.B. (1965). Identity of the site of action of 3', 5' - Adenosine monophosphate and adrenocorticotrophic hormone in corticosteroidogenesis in rat adrenal and beef adrenal cortex slices. Biochemistry 4:462.

Kiviat P.J., Villanueva R., & Markowitz H.M. (1968). SIMSCRIPT II Programming Language. Prentice-Hall.

Koritz S.B. (1962). The effect of calcium ions and freezing on the in-vitro synthesis of pregnenolone by rat adrenal preparations. Biochim. Biophys. Acta 56:63.

Koritz S.B., & Hall P.F. (1964). End product inhibition of the conversion of cholesterol to pregnenolone in an adrenal extract. Biochemistry 3:1298

Koritz S.B., & Kumar A.M. (1970). On the mechanism of action of adrenocorticotrophic hormone. J. Biol. Chem. 245:152.

Koritz S.B., Yun J., & Ferguson J.J. Jr. (1968). Inhibition of Adrenal progesterone biosynthesis by 3', 5'-Cyclic AMP. Endocr. 82:620.

Kraicer J., & Conrad R.G. (1971). Circulating adrenocorticotropin (ACTH) as a controlled variable in the regulation of ACTH secretion. Canadian J. Physiol. & Pharmac. 47:744.

Lefkowitz R.J., Roth J., Pricer W., & Pastan I. (1970). ACTH receptors in the adrenal: specific binding of ACTH-125 I and its relation to adenyl cyclase. Proc. Nat. Acad. Sci. 65:745.

Li C.C., & Urquhart J. (1969). Modeling of Adrenocortical Secretory dynamics. In: Concepts and models of Biomathematics: Simulation techniques and methods. F. Heinmetz & L.D. Cady (Eds.) Marcel-Dekker N.Y.

Liou M.L. (1966). A Novel method of evaluating transient response. Proc. I.E.E.E. 54:20.

Livesey J.H. (1970). A simulation model of calcium Metabolism. Ph.D. Thesis University of Canterbury, Christchurch, New Zealand.

Lowinger J.F. (1969). Magnetic Tape Store and Interface The Recording Format. Third Prof. project report. Electrical Engineering Dept. Univ. of Cant. Christchurch, N.Z.

Lucas H.L. (1964). Stochastic elements in Biological Models; their sources and significance. In: Stochastic Models in Medicine and Biology. J. Gurland (Ed.) The University of Wisconsin Press.

McCausland I. (1969). Introduction to optimal control. Wiley (Publ.)

McKinnon A.E. (1973). Computing Calcium dynamics in Man. Ph.D. Thesis University of Canterbury, Christchurch, N.Z.

McKinnon A.E. (1969). Magnetic Tape Store and Interface Format of Stored Information. Third prof. project report Electrical Engineering Department. Univ. of Cant. Christchurch, N.Z.

Matsuyama H., Ruhmann-Wennhold A., Johnson L.R., & Nelson D.H. (1972). Disappearance rates of Exogenous and Endogenous ACTH from Rat Plasma Measured by Bioassay and Radioimmunoassay. Metabolism 21:30.

Moore W.J. (1966). Physical Chemistry. IV edition. Longmans Green & Co. Ltd (Publ.)

Murphy S.S., Donald R.A., & Nabarro J.D.N. (1969). The half life of Porcine corticotrophin in pigs. Acta Endocr. 61:525.

Noble B. (1964). Numerical Methods: 2 Differences, Integration and Differential Equations Oliver & Boyd (Pub.) NY.

Otterman J. (1960). The properties and methods for computation of exponentially - mapped - past statistical variables. I.R.E. Trans. on Automatic Control. AC-5:11.

Pearlmutter A.F., Rapino E., & Saffran M. (1971). ACTH and cyclic adenine nucleotides do not provoke identical Adrenocortical responses. Endocr. 89:963.

Pritsker A.A.B., & Kiviat P.J. (1969). Simulation with GASP II. Prentice Hall.

Sayers G., Sayers M.A., Tsan-Ying Liang & Long C.N.H. (1946). The effect of pituitary adrenotrophic hormone on the cholesterol and ascorbic acid content of the adrenal of the rat and the guinea pig. Endocr. 38:1.

Schull W.J., & Levin B.R. (1964). Monte Carlo Simulation: Some uses in the genetic study of Primative Man. In. Stochastic Models in Medicine and Biology. J. Gurland (Ed.) The University of Wisconsin Press.

Seelig S., & Sayers G. (1973). Isolated Adrenal cortex cells: ACTH agonists, partial agonists, antagonists; cyclic AMP and corticosterone production. Arch. Biochem. & Bio Phys. 154:230.

Stein M.L., & Rose J. (1960). Changing from Analog to Digital Programming by Digital Techniques. JACM 7:10.

Stokely E.M., & Howard L.L. (1972). Analog computer model for the ACTH - glucocorticoid system. I.E.E.E. Trans. Biomed. Eng. BME 19:13.

Stone D., & Hechter O. (1954). Studies on ACTH action in perfused bovine adrenals: the site of action of ACTH in corticosteroidogenesis. Arch. Biochem. 51,457.

Sulimovici S., & Boyd G.S. (1969). The Δ^5-3β hydroxysteroid dehydrogenase of rat ovarian tissue: the effect of Adenosine 3', 5'-Cyclic Monophosphoric Acid. European J. Biochem. 7:549.

Syn W.M. & Linebarger R.N. (1966). DSL/90 - A digital simulation program for continous system modeling. A.F.I.P.S. conference proceedings vol 28 Spring Joint Computer Conference.

Takebe K., Kunita H., Sakakura M., Huriuchi Y., & Mashimo K. (1971). Suppressive effect of dexamethasone on the rise of CRF activity in the median eminence induced by stress. Endocr. 89:1014.

Thomas G.B. (1964). Calculus 2nd Edition. Addison Wesley (Publ.)

Turner C.D. (1966). General endocrinology 4th Edition. W.B. Saunders Print.

Urquhart J. (1970a). Endocrinology and the systems Paradigm. Behavioral Science 15:57.

Urquhart J. (1970b). Blood-borne signals. The measuring and modelling of Humoral Communication and Control. The Physiologist 13:7.

Urquhart J., & Keller N. (1971). In situ and pilot organ perfusion techniques for the study of metabolic dynamics. Acta Endocr. Supp ## 158.

Urquhart J., Krall R.L., & Li C.C. (1968). Analysis of the Koritz-Hall hypothesis for the regulation of steroidogenesis by ACTH. Endocr. 83:390.

Urquhart J., Krall R.L., & Li C.C. (1970). Adrenocortical secretory function - Communication and control aspects. Automatica 6:193

Urquhart J., & Li C.C. (1968). The dynamics of Adrenocortical secretion. Am. Jnl. Physiol. 214:73.

Urquhart J., & Li C.C. (1969). Dynamic testing and modeling of adrenocortical secretory function. Annals N.Y. Acad. Sci. 156:756.

Urquhart J., Yates F.E., & Herbst A.L. (1959). Hepatic regulation of adrenal cortical function. Endocr. 64:816.

Vecsei P., & Kessler H. (1971). In vivo conversion of radioactive progesterone and corticosterone to adrenal cortical hormones in normal and ACTH treated rats. Acta. Endocr. 68:759.

Wilde D.J. (1964). Optimum seeking methods. Prentice-Hall.

Yates F.E., & Brennan R.D. (1967). Study of the Mammalian adrenal glucocorticoid system by computer simulation.
Conference on hormonal control systems in health and disease, Rancho Santa Fe, Calif., October 1967.
IBM Technical report # 320-3228.

Yates F.E., Brennan R.D., & Urquhart J. (1969). Adrenal glucocorticoid control system. Federation Proceedings 28:71.

Yates F.E., Brennan R.D., Urquhart J., Dallman M.F., Li C.C., & Halpern W. (1968). A continuous system model of adrenocortical function. In: Systems theory and biology. M.D. Mesarovic' (Ed) Springer Verlag Press.

Yates F.E., Leeman S.E., Glenister D.W., & Dallman M.E. (1961) Interaction between plasma corticosterone concentration and adrenocorticotropin releasing stimuli in the rat: evidence for the resit of an endocrine feedback control. Endocr. 69:67.

Yates F.E., Urquhart J. (1962). Control of plasma concentrations of adrenocortical hormones. Physiol. Rev. 42:359.